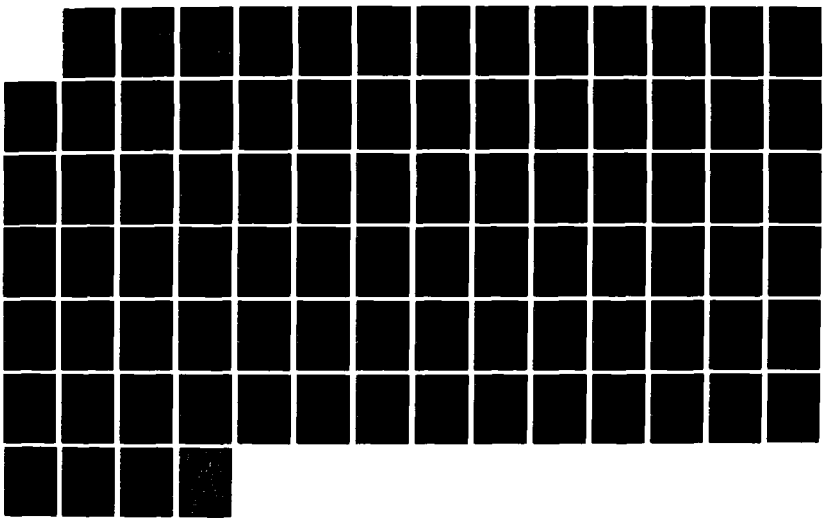
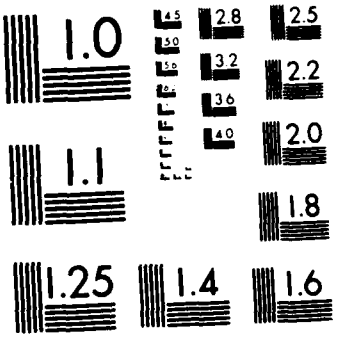
AD-A194 878 DESIGN AND IMPLEMENTATION OF THE COMBINED FEDERAL 1/1

UNCLASSIFIED WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI F L UCMAN DEC 87 AFIT/GCS/ENG/87D-21 F/G 12/7 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3

Design and Implementation
of the
Combined Federal Campaign Collection System

THESIS

Frank L. Ucman
Captain, USAF

AFIT/GCS/ENG/87D-27

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GCS/ENG/87D-27

Design and Implementation of the Combined Federal Campaign Collection System

THESIS

Frank L. Ucman
Captain, USAF

AFIT/GCS/ENG/87D-27

AFIT/GCS/ENG/87D-27

Design and Implementation of the Combined Federal Campaign Collection System

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science (Information Systems)

Frank L. Ucman, B.S.

Captain, USAF

December, 1987

Acknowledgments

I would like to thank my thesis advisor, Capt Mark A. Roth who provided valuable assistance both in editing this document and design methodologies. I also want to thank my thesis committee, Dr. Henry Potoczny and Capt James Howatt, who provided valuable assistance by reviewing my thesis and recommending changes where necessary.

Finally, I want to thank my wife and family for their patience and support throughout this time at AFIT.

Frank L. Ucman

Table of Contents

List of Figures

Abstract

Each year, the Department of Defense and other federal agencies participate in the Combined Federal Campaign (CFC). This campaign consolidates fund raising for many worthwhile charities in a single effort. Computer automation for the annual CFC drive would reduce the number of manhours required to support the annual campaign.

The CFC Collection System developed in this thesis automates many parts of the annual Combined Federal Campaign. The system is menu driven for ease of use and has the capability to establish fund raising goals and produces reports on contributions, agencies recieving money, donating organizations, and historical comparisons. The system reduces the number of man hours required to support the campaign and increases the accuracy and speed of the campaign process.

This thesis effort combined software engineering and database design methods to design and implement the CFC Collection System. The database was designed using the Entity Relationship (E-R) model. The E-R model identified the required data elements and the relationships between elements. The E-R model was then translated into a relational model, producing the tables required for implementation. The implementation of the database files was accomplished using the DBASE III database management system. The application software was designed using the data flow oriented approach to software design. The programs are written using a structured design and implemented in the DBASE III programming language.

Design and Implementation of the Combined Federal Campaign Collection System

I. Introduction

1.1 Background

Each year, DOD and other federal agencies participate in the Combined Federal Campaign (CFC). This campaign consolidates fund raising for many worthwhile charities in a single effort. To support the campaign, each organization provides manpower to oversee and execute the campaign. At the present time the personnel chosen must manually keep track of donations, prepare reports and calculate the donation amounts. [1].

Computer automation for the Combined Federal Campaign would reduce the manhours required to support the campaign. The automation would also provide the ability to compare the present status of campaign donations with anticipated goals as well as results of previous campaigns.

1.2 Problem

Computer automation for the annual Combined Federal Campaign (CFC) drive would reduce the number manhours required to support the annual campaign. The system must :

1. Provide the functions necessary to support the annual CFC campaign as specified in the sponsors requirements definition.
2. Be compatible with the computers and software available on the government small computer contract. This is necessary to preclude the user organization from having to submit special contracts to purchase computer systems or software to use the CFC Collection System.
3. Be user friendly so a person with little computer experience can learn to operate it efficiently with little training.

1.3 Approach

The approach for this project follows a standard design approach. The following are the basis steps for the system design :

1. Requirements Analysis
2. Preliminary design
3. Detailed design
4. Coding
5. Testing

The requirements for this project were established by the sponsor in the thesis proposal; they are outlined in Chapter 2. The requirements specified were refined by information gathered from interviews with the CFC director in Montgomery, AL and the CFC office in Dayton. The requirements analysis step addressed two parts, the data requirements and the software requirements. The data requirements were translated into a data model and the software requirements were the basis for the software design.

The database requirements analysis insured accommodation of all possible CFC charitable agencies, donating organizations, and donations. The data analysis was drawn from previous campaigns. Database entities and relationships were developed using an entity-relationship (E-R) data model to represent relationships between charitable agencies, donating organizations, and funds collected. A relational scheme was developed enabling the storage and retrieval of information without unnecessary redundancy. Functional dependencies and keys were determined for the relational scheme and the best decomposition was chosen to build the actual database for the CFC.

The database management system (DBMS) was chosen from those available under the present small computer contract. The software used to build the system should be readily available to the

personnel who will use it. The software on the small computer contract has already been purchased by many organizations so there is a good possibility that many of the organizations who will use the CFC system would not have to purchase any additional software.

The application software was designed using the data flow oriented approach to software design. The programs are written using a structured design and bottom up coding techniques.

The CFC Collection System has the capability to add, delete or edit agencies as necessary. Donating organization reports are in tabular form. Dispersement of funds and totals for each charitable agency is included in the reporting function of the system. Menu driven programs guide the workers using the system, enabling them to correct errors and review the information as it is keyed in. Backup and recovery procedures are provided automatically where possible and as a utility for the total database is provided.

Software validation was accomplished by running simulated CFC data against the system and comparing the results with predetermined results. The human interface portion of the software was tested by users picked at random who verified ease of use and performance.

1.4 Materials and Equipment

The equipment used for this project was provided by the Air Force Institute of Technology which included a Zenith 248 microcomputer, line printer, associated software and the commercial DBASE III software.

1.5 Sequence of Presentation

The remainder of this thesis is divided into five chapters. Chapter II addresses the requirements the system will have to fulfill. Chapter III describes the methodologies used for designing the CFC Collection System. Chapter IV is divided into two sections addressing the development of the E-R diagram and the implementation of the database tables and describing the menu design and

program flow. Chapter V discusses issues concerning implementation such as special commands of DBASE III and indexing. Constraints of DBASE III which affected the system design are also addressed. The chapter ends with test results. Chapter VI presents a summary of the project and gives somes recommendations for further enchancements.

II. Requirements

The requirements for this project were established by the sponsor in the thesis proposal and are as follows :

1. Establishment of campaign fund raising goals.
2. Distribution and collection of contribution forms. Contributions can be made by cash, check or payroll deduction.
3. Progress reports on organization contributions by amount and participation.
4. Progress reports on contributions by designated charity.
5. Audit functions for cash and contribution form control.
6. Reports showing historical comparisons to previous campaigns.
7. Processing of payroll deductions through JUMPS (Joint Uniform Military Payroll System) and the Civilian Pay system.

The DCS Comptroller, Air University AU/AC, Maxwell AFB, AL agreed to do the payroll interface, processing of payroll deductions through the Joint Uniform Military Payroll System and the Civilian Pay System. The first 6 items were accomplished by the CFC system and the ability to accommodate files from the payroll interface system were included.

Using the requirements specified by the sponsor on the thesis proposal and information gathered from an interview with the CFC director in Montgomery, AL, the following annual CFC campaign sequence of operations was developed.

1. The director sends a questionaire to the participating organizations asking for the rank structure of their personnel. These figures are used to calculate a fund raising goal for each organization based on a suggested gift amount. The suggested gift amount for each pay grade is calculated by taking a percentage of the pay grade amount.

2. The charitable agency catalog is made using the names of the participating charitable agencies. The agencies are grouped by parent agencies and listed by individual agency numbers. The agency provides a twenty five word statement about their purpose and their address.

3. The CFC director contacts the commanders of the participating organizations to get prepared for the campaign. The commander appoints a project officer if one does not already exist. The keyworkers are then chosen by the project officers.

4. The project officers are mailed collection sheets along with donor cards for their organization.

5. The keyworkers contact the organization personnel and receive donations via the donor cards. The donations can by made by cash, check or allotment. The donation can be designated to one or several charitable agencies. The agency can also be specified by writing in the name and address if it is not in the CFC campaign catalog.

6. The donor cards are returned and totaled for the organization by the project officer. The project officer returns the received donor cards and the total sheet to the CFC campaign director. Donor cards are checked for accuracy and corrected if necessary.

7. The CFC director totals all organization's contributions to determine the overall campaign status, briefs the commanders, and issues news updates for the campaign.

8. During the campaign awards are given to people who have given enough to receive an award. Typically, the awards are pins, pendants, or letters of acknowledgement.

9. At the end of the campaign, the organization totals are verified for accuracy and corrected if necessary. The distribution of funds to agencies is figured by taking the cost of the campaign and dividing it between the receiving agencies. The funds are then distributed to the receiving agencies.

Chapter III discusses the database and software methodologies used for the CFC Collection System.

III. Database and Software Design Methodologies

The organization of information into a usable format, used to build the CFC Collection System, is accomplished by software called a Database Management System (DBMS).

A database is a collection of related information stored in one place and organized so that the information can be retrieved quickly and efficiently [2]. The information is grouped into records. Each record is composed of fields, which contain data or values. The DBMS is the software used to manage this information.

The following are some of the ways the DBMS manages information [2]:

1. Organizes and stores information.
2. Finds information quickly.
3. Reorganizes information to suit needs.
4. Displays information and prints reports.

After the design of the database is complete, attention is directed to the design of the software using the database as a tool. By integrating an established software design method with the database, an efficient and maintainable system is provided.

In this chapter database modeling and software design will be discussed.

3.1 Database Modeling

The database is modeled using an Entity-Relationship model. The Entity-Relationship (E-R) data model is based on a perception of the real world based on a set of basic objects. These objects are called entities. The ways in which these objects address each other are called relationships [3]. This allows the specification of an enterprise scheme to represent the overall logical structure of the database.

An entity is an object which exists and can be distinguished from other objects. An example would be a person entity, such as John Doe with a social security number of 123-45-6789 who can be uniquely identified from another person. The attributes that represent the entity could be name and social security number.

The entities to be accessed in the database are made distinct by determining keys. Keys are expressed in terms of the entities' attributes. A key which is assigned to an entity set which will uniquely identify each entity is called a superkey. The superkey may be one or more of the attributes in the entity. An example would be the combination of a person's social security number and name together. The superkeys are reduced to a minimal set of attributes called candidate keys which will uniquely identify the entity. The candidate key that will be used as the primary means of identifying the entity is chosen as the primary key. In the person entity the social security number would be the primary key to identify the person entity. An entity which has a primary key is termed a strong entity.

In case there exists an entity in which a primary key can not be formed from the attributes, it is referred to as a weak entity since its existence is dependent on a strong entity. The key for a weak entity is a combination of the primary key of the strong entity and one or more attributes of the weak entity called a discriminator.

The overall logical structure of a database can be represented graphically by the E-R diagram using the following components [4] :

1. Rectangles, which represent entity sets.
2. Ellipses, which represent attributes.
3. Diamonds, which represent relationship sets.
4. Lines, which link attributes to entity sets and entity sets to relationship sets.

The E-R diagram defines the constraints to which the database contents must conform. Mapping cardinalities express the number of entities to which other entities can be associated .

The relationships for the E-R diagram are as follows :

1. One-to-one : one entity relates to one entity.
2. One-to-many : one entity relates to many entities.
3. Many-to-one : many entities relate to one entity.
4. Many-to-may : many entities relate to many entities [4].

Existence dependencies form a class of constraints specifying a dominant entity and the subordinate entity. An example of this, (Figure 1), would be the existence of a "works for" relationship between a person and an organization. If a person "works for" an organization both the person and the organization have to exist for the "works for" relationship to exist. The "works for" relationship is subordinate to the dominant entities, person and organization. If the person entity or the organization entity is deleted the works for entity is deleted also. If the "works for" relation has data unique to the relationship, the data would be stored in a file that represents the relation with a field common to person and a field common to organization.

Figure 1. E-R Diagram of Person works for an Organization

The database represented by the E-R diagram can be reduced to a collection of tables. Then each entity and relationship will have a corresponding table. Figure 2 is the table for the file which holds the personnel information. The tables for all the files in the database for the CFC system are located in Appendix B.

Attribute	Type	Length
SSAN	Numeric	9
LNAME	Character	10
FINIT	Character	1
TYP	Character	1
SERVICE	Character	5
GRADE	Character	5
ORG_CODE	Numeric	3
SUB_CODE	Character	2
PHONE_NO	Character	14
KEYWORKER	Logical	1
CARD_NO	Numeric	5

Figure 2. Personnel Table

3.2 Software Design

Once the software requirements have been established and the data model designed, the development phase of the software is begun. The development phase has four distinct steps [3] :

1. Preliminary design
2. Detailed design
3. Coding
4. Testing

Software design is a process in which the requirements are translated into a representation of software. The design should meet the following characteristics [3] :

1. Heirarchical organization which makes intelligent use of control among elements of software.
2. Modular in which the software is logically partitioned into parts that perform specific functions.
3. Modules should have independent functional characteristics.
4. Readable design that is driven by the requirements.

The design method for this system will be a data flow-oriented design method. The objective of this method is to provide a systematic approach for the development of software. The data flow-oriented design is based on the strategy of transform analysis in which we base the design on the transformations of the data as it passes through the system.

IV. CFC System Design

This chapter covers database design and program flow. The database design is an explanation of how the E-R diagram was made into tables. The tables represent the actual database files. Appendix A is a description of the files for the CFC Collection System. The software design section give the flow of the CFC Collection System and examples of the menus.

4.1 CFC Database Design

The database is modeled using an E-R diagram. The E-R diagram was developed by using the standard CFC donation card, and current CFC reports and by interviewing the personnel at the CFC office in Dayton, OH and the CFC director in Montgomery, AL. This section discusses how the E-R model was developed and how the translation of the E-R model into tables was accomplished.

4.1.1 Discussion of E-R model. The E-R diagram in Figure 3 is a model of the CFC database. The following section describes how the relationships were developed.

The CFC campaign is driven by the donor card and has participating organizations with personnel who give donations to charitable agencies. The entities identified were persons, organizations, agencies and the donor card.

The participating organizations have personnel working for them which is modeled by the "works for" relation. The person working for the organization maybe a keyworker and the organization could have zero to many keyworkers. The "isa keyworker" and the relation "cfc wkr" model this. The organizations usually are in a hierarchy and this is depicted by the relation "is child ".

The charitable agencies which receive the donations are modeled by the agency entity which also can be grouped by the "is under" relation.

The donor card has information on it that may or may not be part of the entities discussed thus far. This is why the donor card has the following relations. P-gift ties the donor card to a

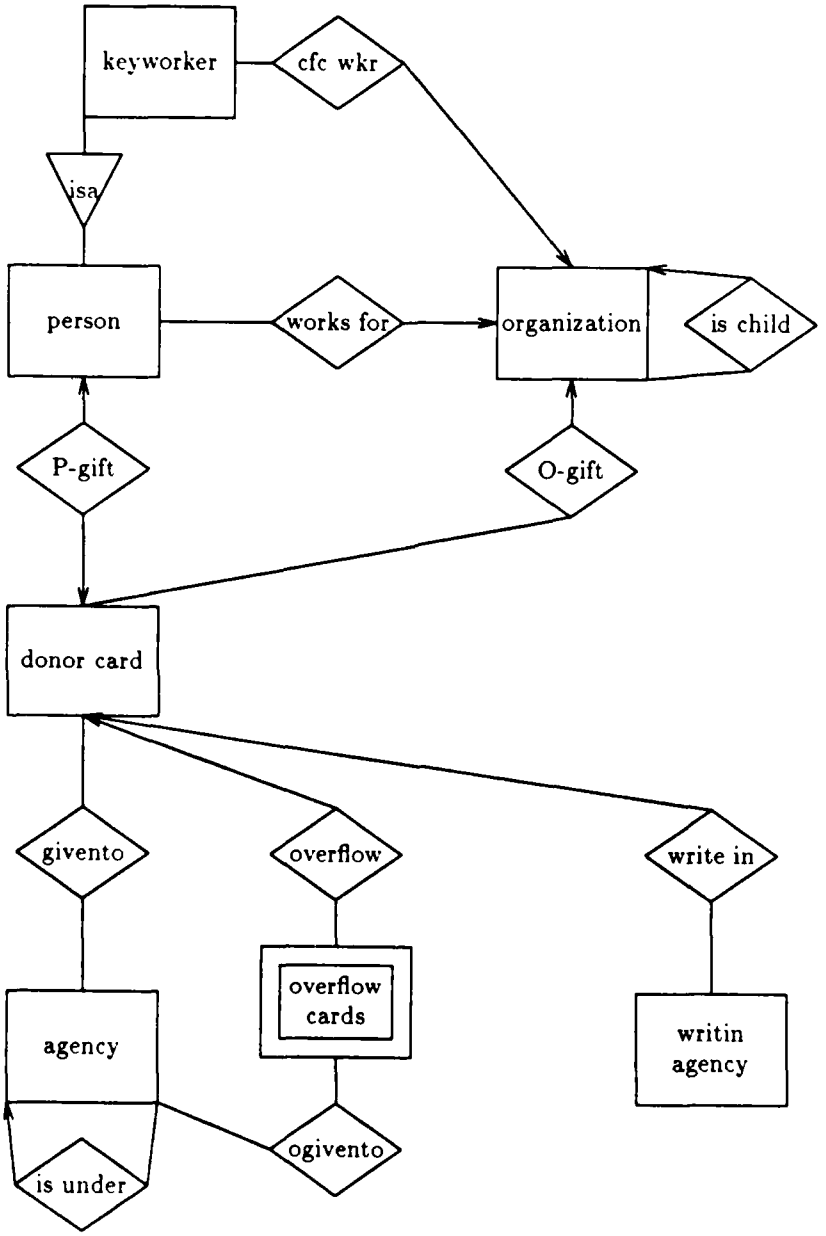


Figure 3. E-R Diagram of CFC Database

person, if one is named on the donor card. The donor cards are all given by some organization which has to know the percent participation of its personnel, so donor card is related to organization by O-gift.

The donor card has five entries on it to designate agencies for specified amounts. Donors can specify any number of agencies, or write in an agency that is not in the catalog. Amounts not designated are divided among certain participating charitable agencies. To accommodate this the donor card was divided into the following relations. Givento relates donor cards to agencies, the implementation is one card to five agencies because of the card size. The overflow cards entity is referred to as a weak entity since it can only exist when the donor card has exceeded the five designated charities. The relation "overflow" relating the donor card to an overflow card and the overflow card to the agencies via the "ogivento" relation enables the designation of more charitiable agencies for the donor card. The limit for the overflow card is five on the implementation because of the card size.

There maybe cases where the donor would like to participate in the campaign but does not want to give to any agency in the catalog. The "write in" relation accommodates this situation enabling the donor card to be related to a write in agency.

4.1.2 Translation into Tables. Once the E-R diagram was completed, the data elements were made into tables which represent the database entities. The data elements are defined in two groups, the donation card and support data.

The support data was divided into four tables. Three of these are in the E-R diagram. The suggested gift, organization,personnel, and agency tables are the support tables.

1. The Suggested Gift table holds suggested gift information and is used for goal projections and error checking donor cards for proper pay grades when an allotment is used for the donation

   The attributes are :

   key: PAY_GRADE

   SUGGESTED_GIFT

2. The Organization table holds information for organizations that have participated in the CFC campaign. The attributes chosen support goal projections, reporting on organizations, printing labels and building an organizational hierarchary number of persons and goal.

   The attributes are :

   key : ORGANIZATION_CODE, SUB_CODE

   ORGANIZATION_NAME, ADDRESS, CITY, STATE, ZIP,

   COMMANDER, COMMANDERS_PHONE,

   PROJECT_OFFICER, PROJECT_OFFICERS_PHONE,

   NO_PERSON, GOAL, PROPOSED_GOAL,

   HIERARCHICAL_NO_PERSON, HIERARCHICAL_GOAL,

   PARENT_ORG_CODE, PARENT_SUB_CODE

   ORGANIZATION_CODE and SUB_CODE builds the relationship Works for and O-gift by placing ORGANIZATION_CODE and SUB_CODE in the Doncard and Person tables.

   PARENT_ORGANIZATION_CODE, PARENT_SUB_CODE are used for the relationship Is child to determine the parent organization for reporting and for calculating goals for the organizational hierarchy.

3. The Person table holds information on personnel. Personnel information is needed when the donor makes a donation by allotment or wants to have his or her donation recorded. This would be used for an award program which sends out acknowledgements of the donations which exceed certain amounts. The error checking procedures use this table to check for duplicate donations per person. Reporting of donations for military personnel and civilians also uses personnel information.

   The attributes are :

   key : SSAN

   LAST_NAME, FIRST_INITIAL,

   TYPE(military or civilian), SERVICE, PAY_GRADE,PHONE_NO,

   CARD_NO, ORGANIZATION_CODE, SUB_CODE, KEYWORKER

   The P-gift relationship relates a person to a donor card by CARD_NO on a one-to-one basis. The O-gift relationship uses ORGANIZATION_CODE and SUB_CODE to relate a donor card to an organization on a many-to-one basis. KEYWORKER identifies whether or not the person is a campaign keyworker, implemening the "isa keyworker" relationship.

4. The Agency table contains valid agencies for the CFC campaign. The agency table is used for error checking donor cards for valid agencies when the donor designates specific agencies to receive all or part or his or her donation. The agency table is also needed for reporting, distributing funds at the end of the campaign and making the CFC catalog of charitable agencies for the annual campaign.

   The attributes are :

   key : AGENCY_NO

   AGENCY_NAME, ADDRESS, CITY, STATE, ZIP, TELEPHONE, DESCRIPTION,

   PARENT_AGENCY_NO

   Agency is related to Doncard via Givento and Ogivento using AGENCY_NO and CARD_NO.

The donation card data was divided into four tables to get maximum use of space and to give flexibility to donation entry possibilities. The explaination of the donation card tables is as follows:

1. Doncard contains the attributes from the donor card that are common to every donor card.

   The attributes are :

   key : CARD_NO

   TYPE (military or civilian), PAY_GRADE, GIFT_LEVEL, WEEK, CASH, ALLOTMENT, TOTAL, SSAN, ORGANIZATION_CODE, SUB_CODE, WRITIN, DESIGNATED

   ORG_CODE and SUB_CODE makes the O-gift relation relating Doncard to Organ on a many-to-one basis. CARD_NO is used in several relations. P-gift relates Doncard to Person on a one-to-one basis. Writin relates Doncard to Writin Agency on a one-to-many basis. Agency is related to Doncard via Givento and Ogivento by on a many-to-many basis. The common fields are CARD_NO and AGENCY_NO.

2. The Givento table holds five agencies and an amount for each agency. Givento is used when an agency is designated on the donor card and is utilized fairly often. The attribute OVERFLOW is to indicate whether or not overflow cards for this donor exist.

   The attributes are :

   key : CARD_NO

   AGENCY_1, AMOUNT_1, AGENCY_2, AMOUNT_2, AGENCY_3, AMOUNT_3, AGENCY_4, AMOUNT_4, AGENCY_5, AMOUNT_5, OVERFLOW

   The Givento relation relates Agency to Doncard using AGENCY_NO and CARD_NO on a many-to-many basis.

3. The Ogivento table holds five agencies and an amount for each. This file is used when a donor wants to specify more than five agencies on a donation card. These donation cards are referred to as overflow cards. This file is rarely used. Most donors do not designate more than five agencies. The extra file saves space in the database for the donor card entry, as opposed to having extra fields in Givento which would be left empty most of the time.

   The attributes are :

   key : CARD_NO

   AGENCY_1, AMOUNT_1, AGENCY_2, AMOUNT_2, AGENCY_3, AMOUNT_3,

   AGENCY_4, AMOUNT_4, AGENCY_5, AMOUNT_5, OVFLO_NO

   The Ogivento relation relates Agency to Doncard using AGENCY_NO and CARD_NO on a many-to-many basis.

4. The Write in agency table allows a donor to specify an agency that is not in the CFC catalog. People occasionally write in the agency they want their donation to go to. This table enables the system to handle the write in agencies and also to report on them so they can be added to the CFC catalog for the next campaign.

   The attributes are :

   key : CARD_NO

   AGENCY_NAME, ADDRESS, CITY, STATE, ZIP,

   ANNUAL_AMOUNT, WRITE_IN_NO

   The writin relation uses CARD_NO to relate a write in agency to a donor card on a many-to-one basis allowing a donor to specify as many write in donations as he or she wants.

4.2 CFC System Application Design

4.2.1 Preliminary design. Using the requirements definition stated in Chapter II and knowledge gained from interviews with the CFC offices in Dayton, OH and Montgomery, AL functional charts of the CFC Collection System were developed. The functional descriptions of the software modules and the data elements were then defined.

4.2.2 Detailed design. The detailed design began with the layout of the menu system and the designing of the menus and screen formats necessary to input data. A major consideration was error handling techniques for donor card entry along with the options for loading and unloading donations to and from external files. The system has upward mobility of data which means many systems can be used by keyworkers, the data can be copied to diskette, uploaded and merged into one central system. This feature will enable many people to share the workload by having more than one copy of the system to use. The CFC Collection System is designed with several report features to enable information to be extracted from the database in many forms. The utility features have uses other than just the annual campaign. The CFC office can use the Agency Utilities to keep agency information and perform reports and updates on it for the next annual campaign. The personnel and organization utilities can be used for organization reporting and office use.

The program flow will be described using a flow chart to introduce a group of menus followed by a description of the menu options.

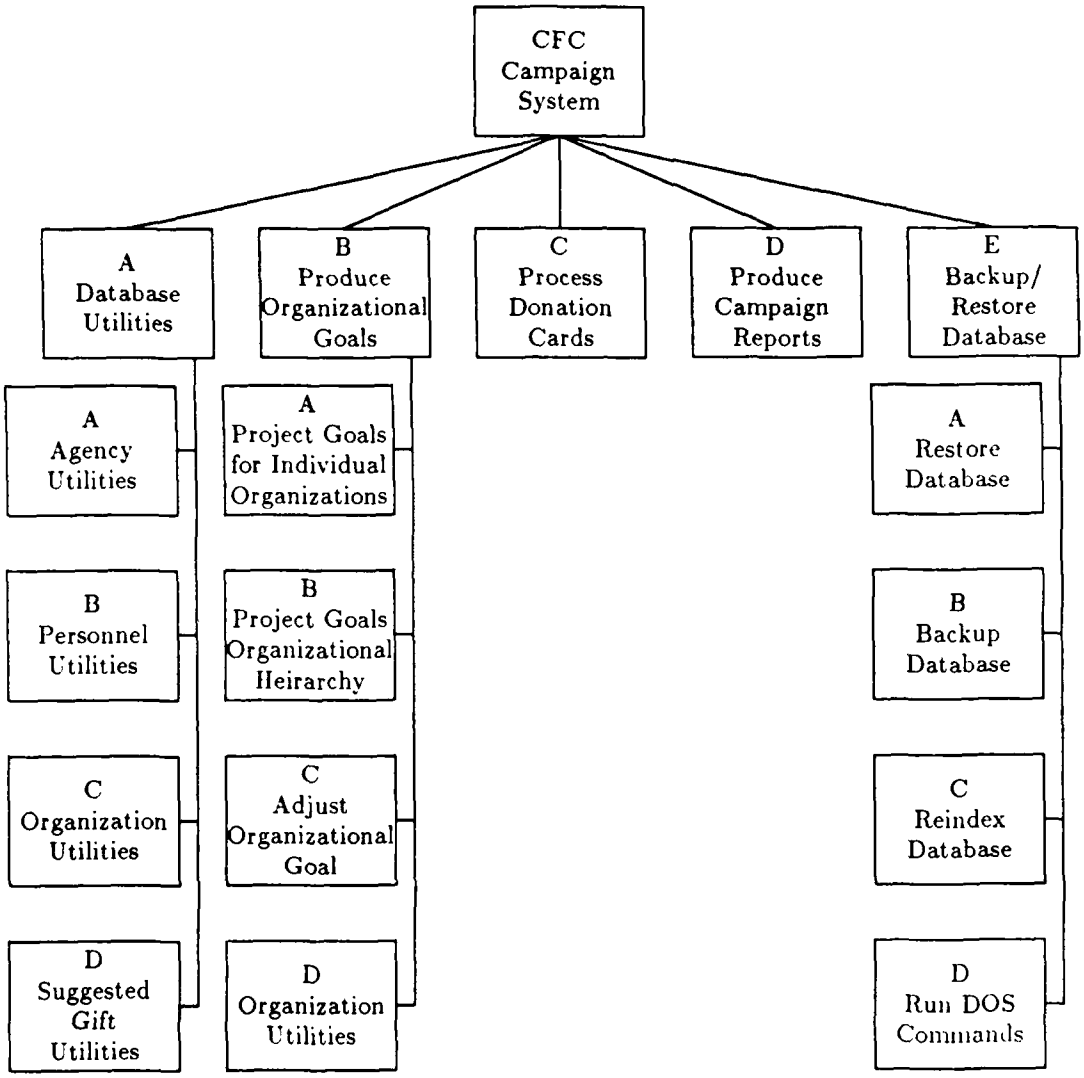


Figure 4. CFC Campaign System, Main Menu

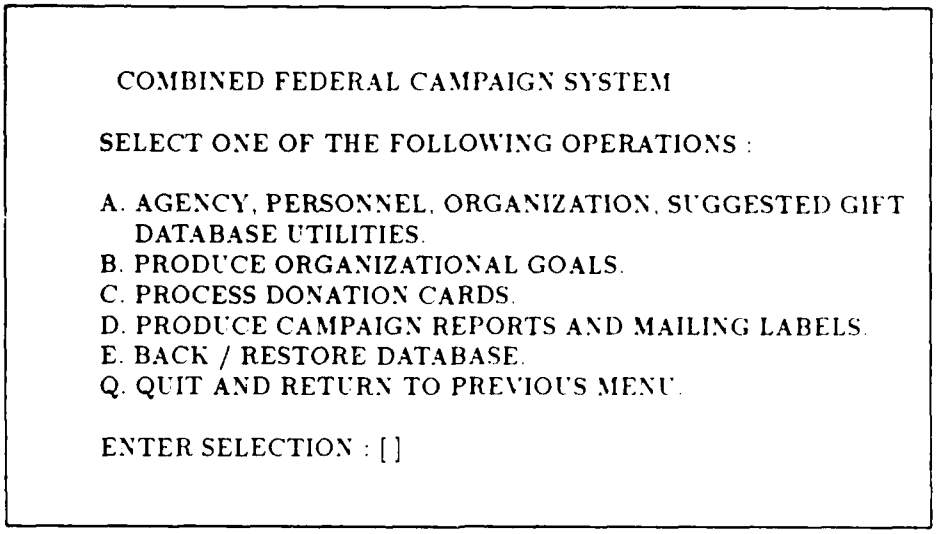


Figure 5. Main Menu : CFC Campaign System

A. Database Utilities : The database utilities handle all the functions necessary for file maintenance. The database utilities are divided into four subfunctions to handle the agency, personnel, organization and suggift files. Option C, Process Donation Cards, works the same as Option A, Database Utilities. The example in this manual is the Agency Utilities option.

B. Produce Organization Goals : This option is used to calculate and project goals for organizations.

C. Process Donation Cards : The donation utilities handle all the functions necessary for processing of donation cards for an annual campaign.

D. Produce Campaign Reports and Mailing Labels: Enables reporting of campaign status, builds summary files, historical reports, project officer collection sheets and makes mailing labels for organizations.

E. Backup/Restore : Recovery procedures for the database and associated files.

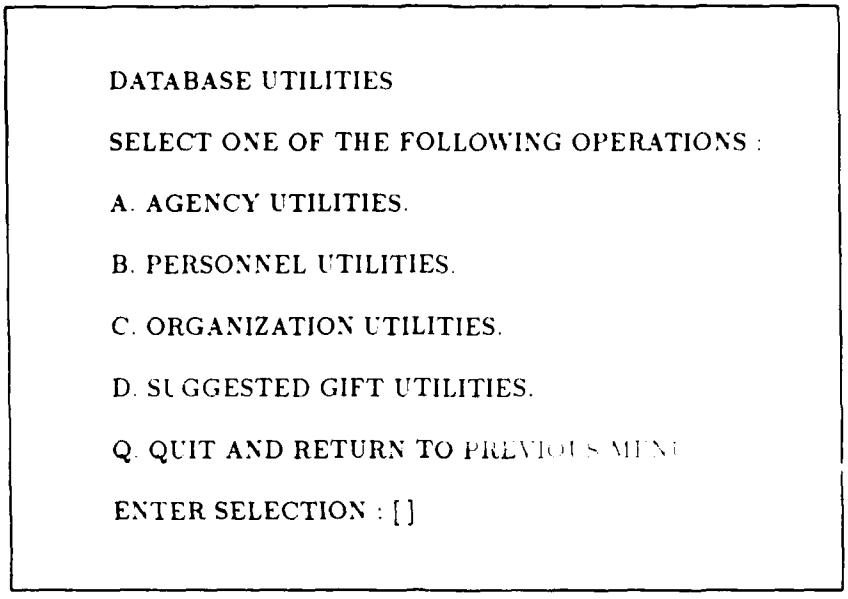


Figure 6. Option A : Database Utilities

There are four Database Utility functions for the system. For example there are Agency Utilities for modifying or reporting on the agency database. The agency numbers have to be loaded prior to running the campaign. The donor cards are checked for valid agencies before they are written to the database. The Personnel Utilities are for personnel data; this may be loaded prior to the campaign, but is not necessary. Personnel not in the database are added when the donation card is entered. Organization Utilities control the organization database. The organizations must be loaded prior to the campaign as the donor cards are checked for valid organizations. Suggested Gift Utilities load the suggested gift file. The suggested gift file is used during the campaign to error check pay grades and is also used to project organizational goals. Option C, from the main menu, Process Donation Cards operates in the same manner as the Database Utilities. The description for the database utilities in the document is Option A, Agency Utilities.

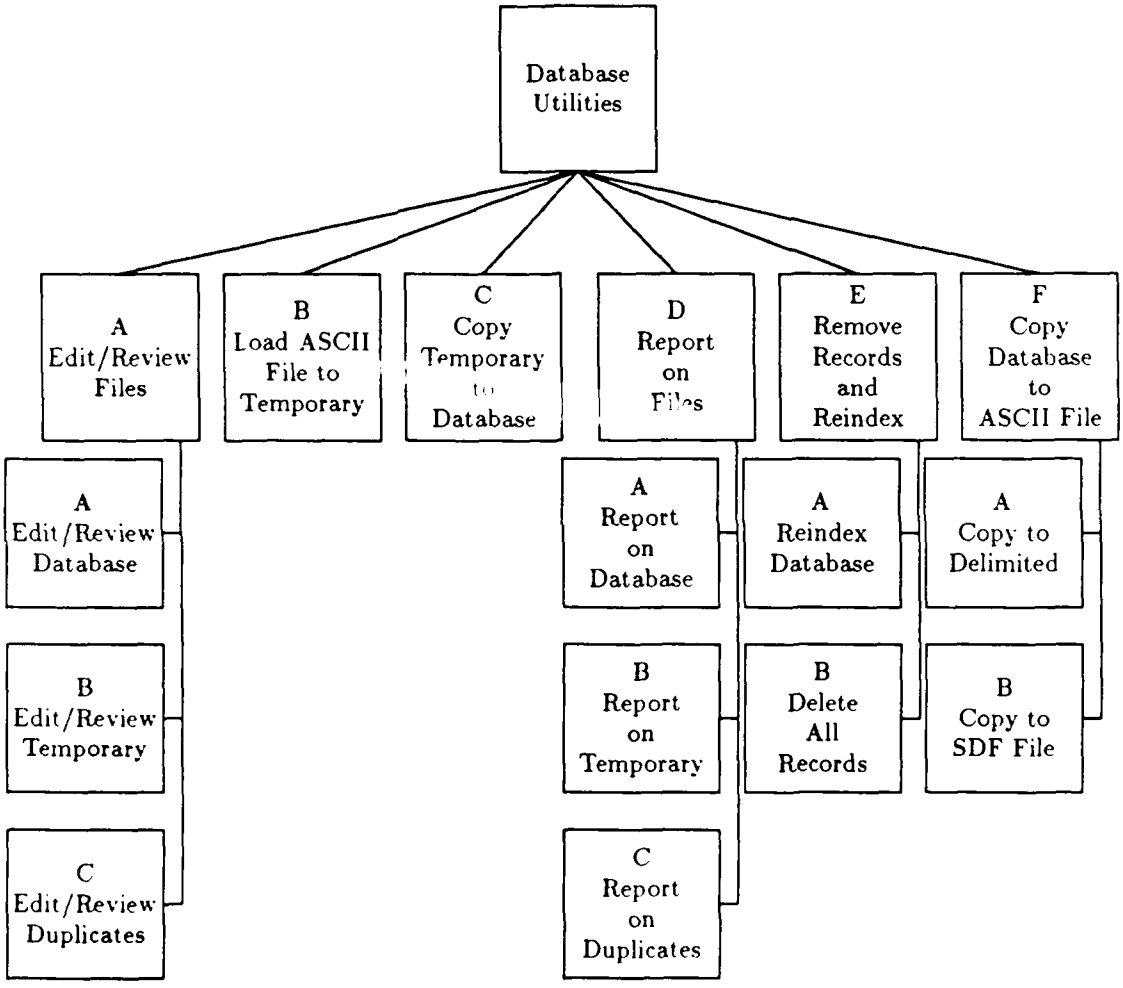


Figure 7. Database Utilities : Agency, Personnel, Organization, Suggest Gift, Donation

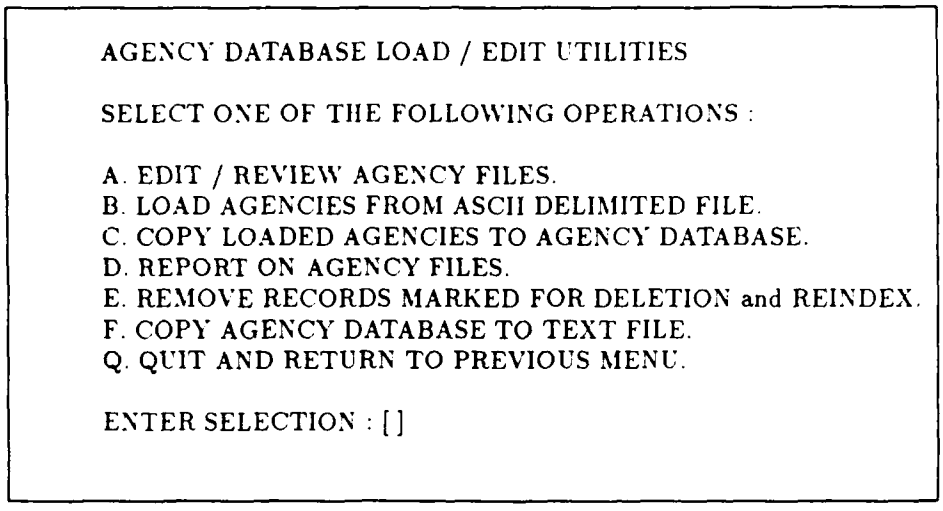


Figure 8. Database Load / Edit Utilities

A. Edit/review files : Three editor options are associated with each database. The editors allow the addition of new records in the actual database. Modification and deletion of records is available on all editors. The three editors edit the temporary file which is for batch processing, a duplicate for errors when batch processing and the actual database.

B. Load ASCII file to temporary database : This allows the copying of records into the temporary database from an ASCII file and is used in conjunction with option C. This is used to unload or reload the database file and allows data to be moved from one system to another.

C. Copy temporary to database : Copies a temporary database to the actual database and performs error checking. Any record with errors are written to the duplicate file and good records are written to the actual database.

D. Report on files : Three reports are associated for each database one for the actual file, one for the temporary, and one for the duplicate.

E. Remove records and reindex : Two options are available. The first option is to remove records marked for deletion and reindex the files. This is used when a possible error has

occured when using the system and the utility no longer works properly. An example of a possible error would be a power flux or turning the system off while the system is operating on a database file. Rebuilding the index in this case will usually correct the problem. The second option is to delete all records in the file and reindex, this option is used when the database files need to be initialized such as for a new campaign.

F. Copy database to ASCII file : This is used to save a database to an ASCII file for backup purposes or to upload to another CFC Collection System.

```
AGENCY DATABASE EDITOR

SELECT ONE OF THE FOLLOWING OPERATIONS :

A. EDIT / REVIEW AGENCY FILE.
B. EDIT / REVIEW AGENCIES LOADED FROM ASCII DELIMITED FILE.
C. EDIT / REVIEW DUPLICATES FROM LAST LOAD TO DATABASE.
Q. QUIT AND RETURN TO PREVIOUS MENU.

ENTER SELECTION : [ ]
```

Figure 9. Option Database Utilities.A : Edit/Review Files

A. Edit/Review database : Allows editing of the actual database file or files associated with the utility.

B. Edit/Review database Loaded from ASCII Delimited File: Allows editing of the temporary database associated with the utility.

C. Edit/Review duplicates from last Load to Database : Allows editing of the duplicates or errors from copying the temporary database to the actual database associated with the utility.

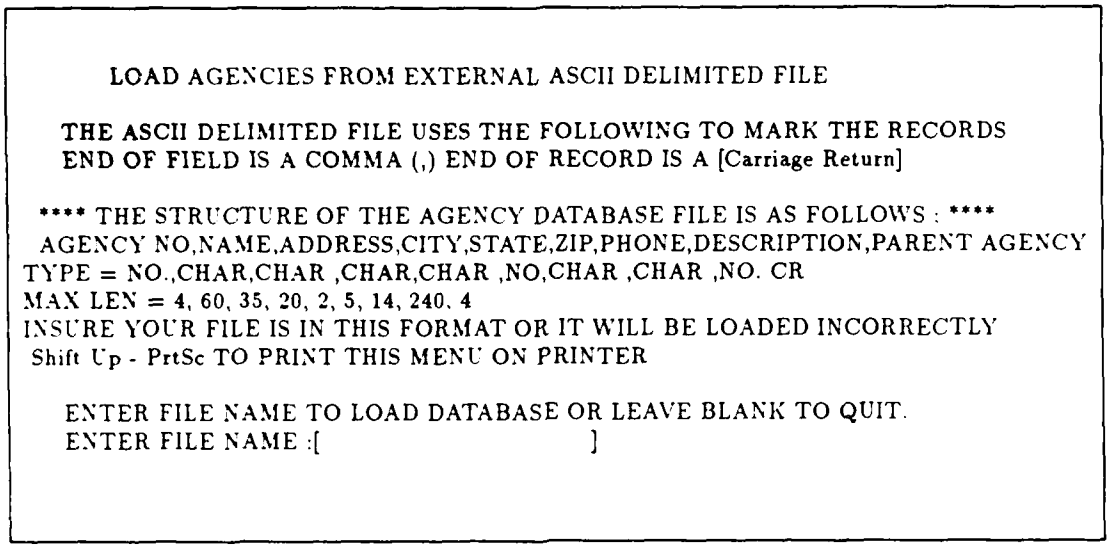


Figure 10. Option Database Utilities.B : Load ASCII File to Temporary

Figure 10 shows the menu for Load ASCII file to temporary database. The menu gives the format for the ASCII file to be loaded the fields are separated by a comma (,) and the records by the carriage control character. This option loads an ASCII file to the temporary database file for review. The temporary file is then loaded to the actual database file using Option C. from the utility menu. Error checking is performed on the data before it goes into the database. The records without errors are stored in the database. The records that have errors are written to the duplicates file and can be reported on or viewed by the editor. The duplicates file for the Process Donation Cards has the field or fields with errors in them are marked with asterisks (*) and an error code number. The temporary database file can be edited. Records with errors can be corrected or deleted in the temporary file and recopied to the database.

The most common errors when loading the ASCII file will be the comma (,) not separating the fields properly, numbers where characters are supposed to be, and no carriage return at the end of the record.

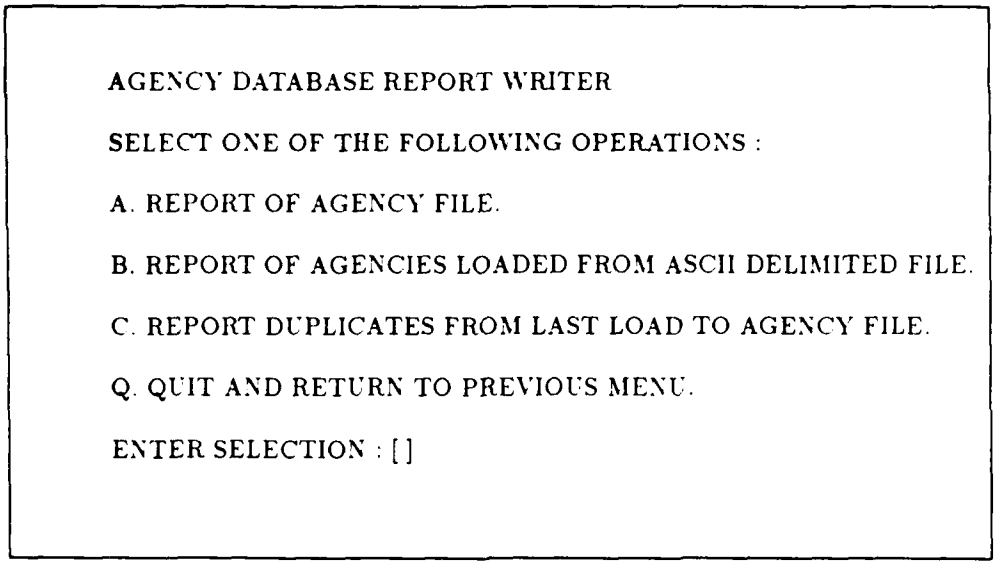


Figure 11. Option Database Utilities.D : Report on Files

A. Report of database File: Produces a report of the actual database.

B. Report of database Loaded from ASCII Delimited File : Produces a report of the temporary database.

C. Report of Duplicates From Last from Load to database : Produces a report of the donation database which is built from copying the temporary database to the actual database. The Process Donation Cards, option D from the main menu, has an option D to show a screen of possible error codes with error descriptions. The duplicates in this case are printed with the donor card and an error description card with asterisks (*) and where possible a number to indicate the type of error in the field with the error.

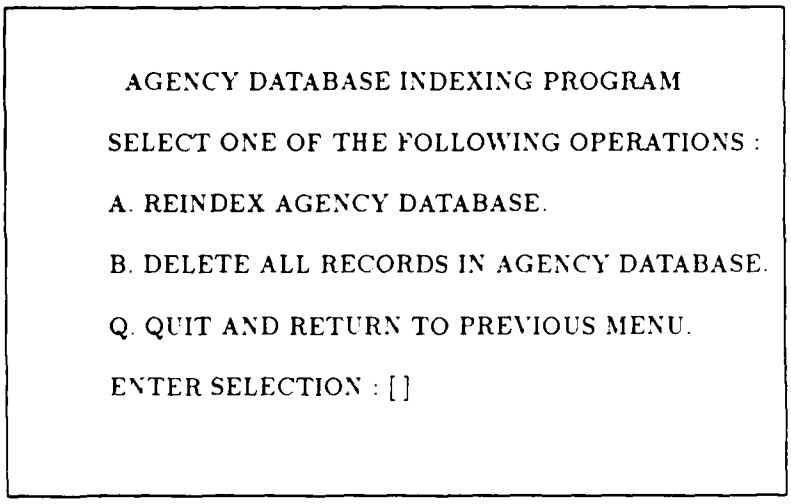


Figure 12. Option Database Utilities.E : Remove Records and Reindex

A. Reindex Database : Reindexes and removes deleted records from the database associated with the utility.

B. Delete All Records : Removes all records from the database associated with the utility.

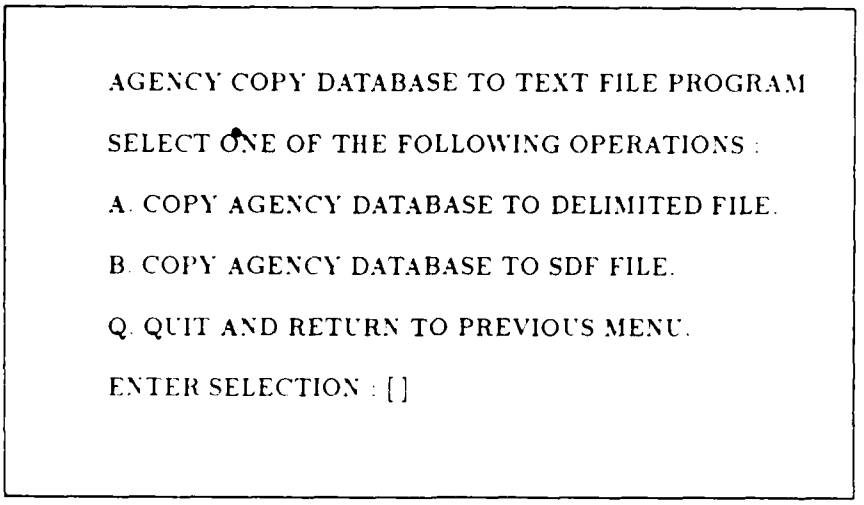


Figure 13. Option Database Utilities.F : Copy Database to ASCII File

A. Copy database to Delimited File : This is used to save the database to an ACSII delimited file. The delimited file can be reloaded to the same CFC System or different CFC System. The delimited file uses the comma (,) to separate fields in the records and the carriage return to mark end of record.

B. Copy database to SDF File : This is used to save the database to Standard Data Format (SDF) file. The SDF format puts the data into a file in record length format. The SDF file can not be read back into the system.

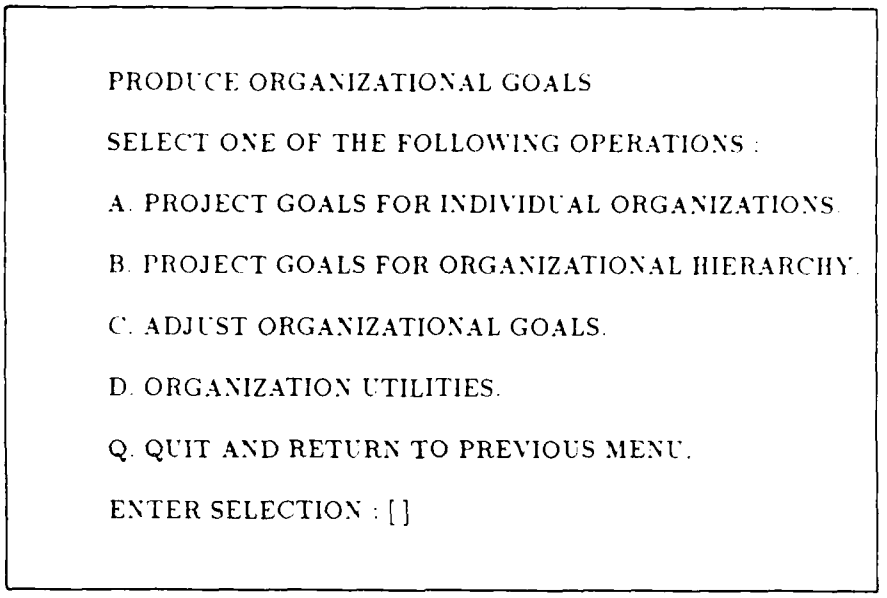


Figure 14. Option B : Build Organizational Goals

A. Project Goals for Individual Organizations : The goals for the individual organizations are calculated by multiplying the suggested gifts by the number of individuals in the organization by pay grades.

B. Project Goals for Organizational Hierarchy : This projects the campaign goals for all organizations in the database by summing the projected goals for the individual organizations in a hierarchy to build the totals for the command organizations and the database as a whole.

C. Adjust Organization Goals : This option adjusts all the organization goals in the database to meet a specific amount.

D. Organization Utilities : These are the utility programs are for the organization database.

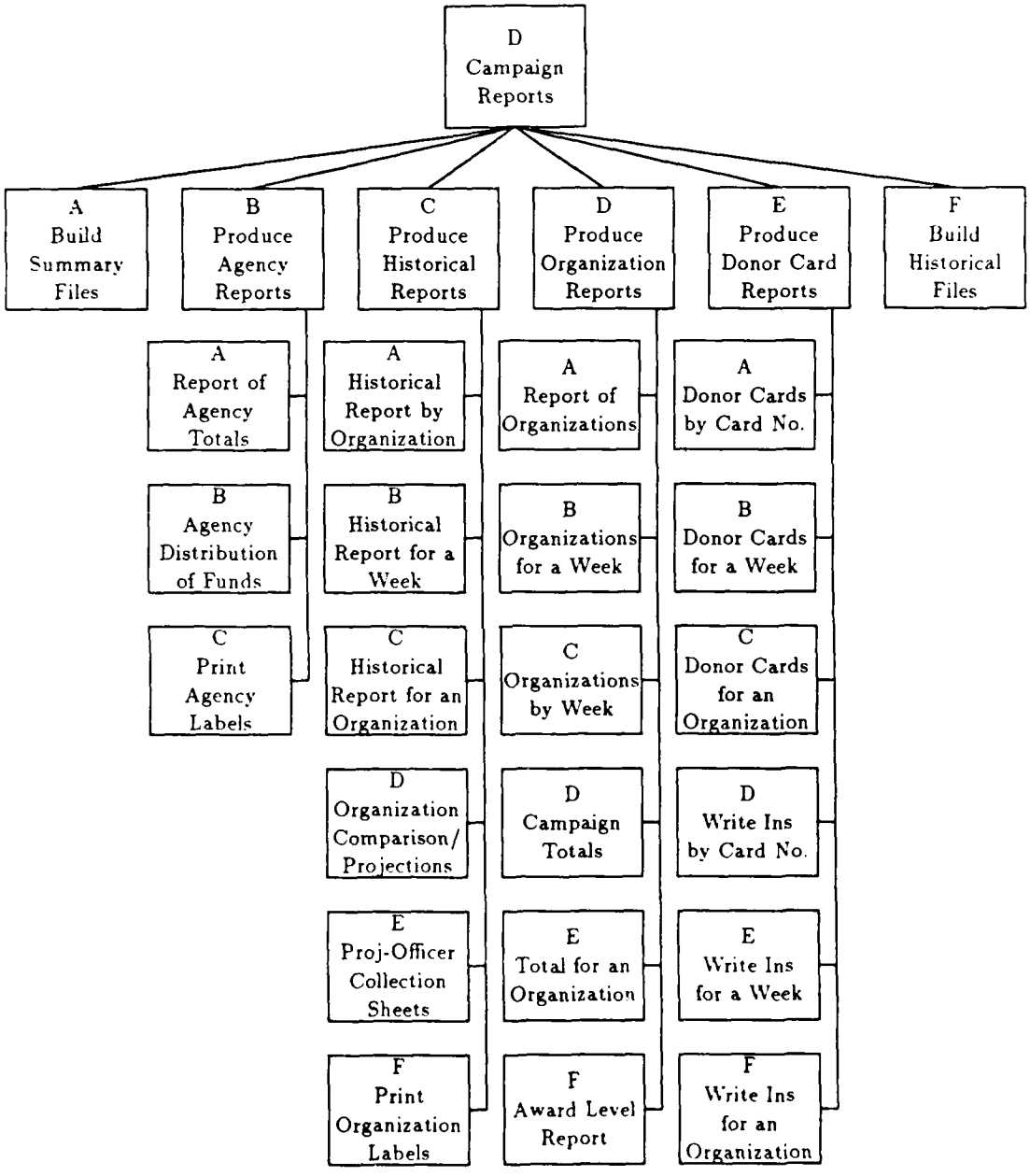


Figure 15. Campaign Reports:D

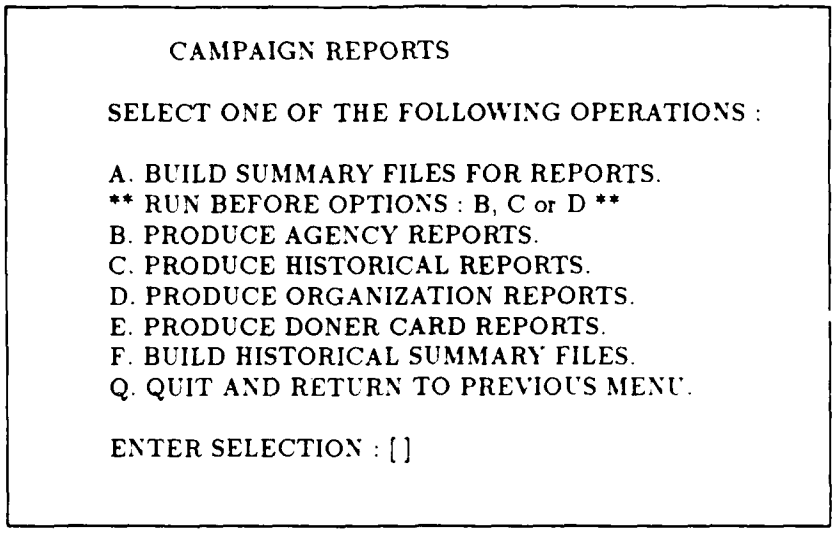


Figure 16. Option D : Campaign Reports

A. Build Summary Files : This option builds summary files for the organization summary reports and the agency summary reports. The reports in Options B,C and D use the files built in this option to make reports.

B. Produce Agency Reports : Produces agency reports giving totals of donations to each designated agency.

C. Produce Historical Reports : Produces comparison reports of present campaign status to a previous campaign which was build with Option F in a previous year.

D. Produce Organization Reports : Produces reports of organizations for all organizations, for a single week, for a particular organization grouped by week. Totals for the campaign are reported under this option.

E. Produce Donor Card Reports : Reports of donor cards for the campaign by card number, for a particular week and for an organization for both regular donations and write in donations.

F. Build Historical files : This builds the organization total file for a year and is used for the historical comparison report.

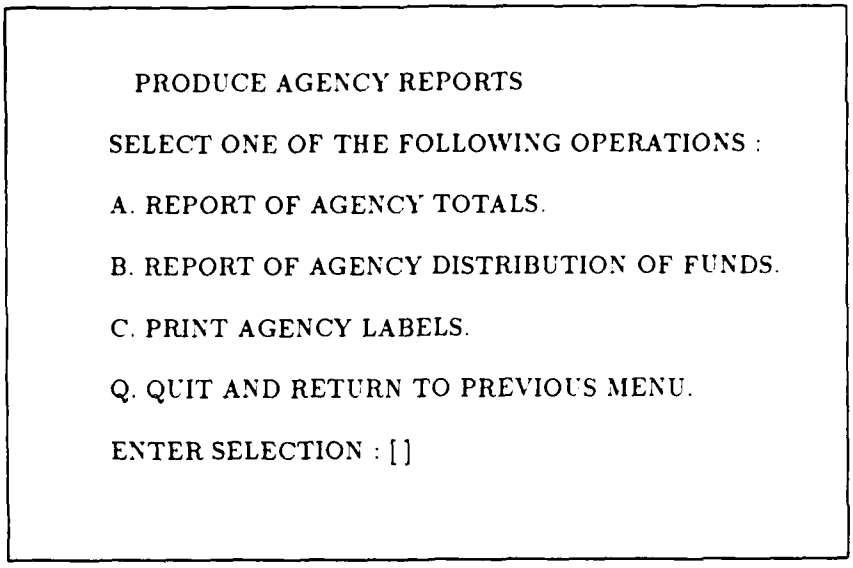


Figure 17. Option D.B : Produce Agency Reports

A. Report of Agency Totals : Produces a report of the total designated for each week and the total to date for each designated agency.

B. Report of Agency Distribution of Funds : Produces a report for the distribution of funds designated to specific agencies.

C. Print Agency Labels : Prints mailing labels for agency mail outs.

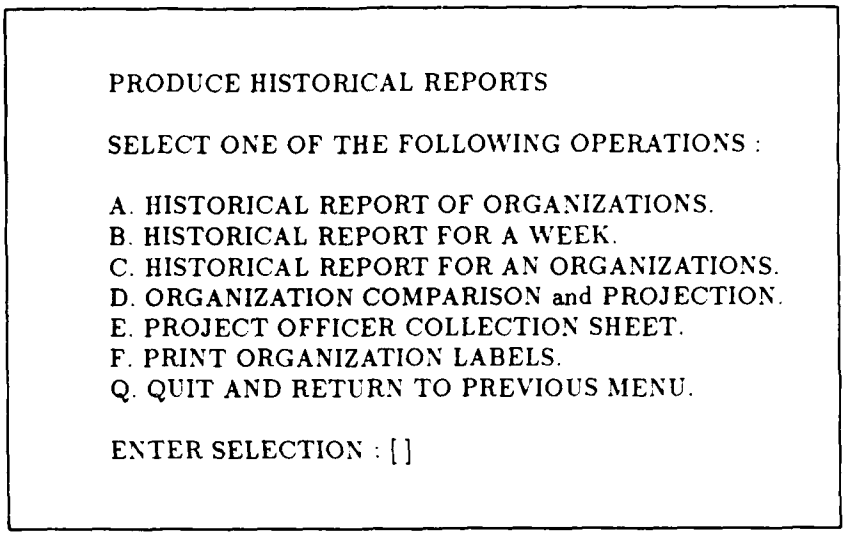


Figure 18. Option D.C : Produce Historical Reports

A. Historical Report of Organizations : Produces a report of all organizations in comparison to a previous year.

B. Historical Report for a Week : Produces a report of all organizations in comparison to a previous year for a particular week.

C. Historical Report for an Organization : Produces a report for a particular organization in comparison to a previous year.

D. Organization Comparison and Projection : Produces a report of all organizations with projections for the campaign in comparison to a previous year.

E. Project Officer Collection Sheet : Produces the collection sheet for an organization project officer.

F. Print Organization Labels : Makes mailing labels for organization mailouts.

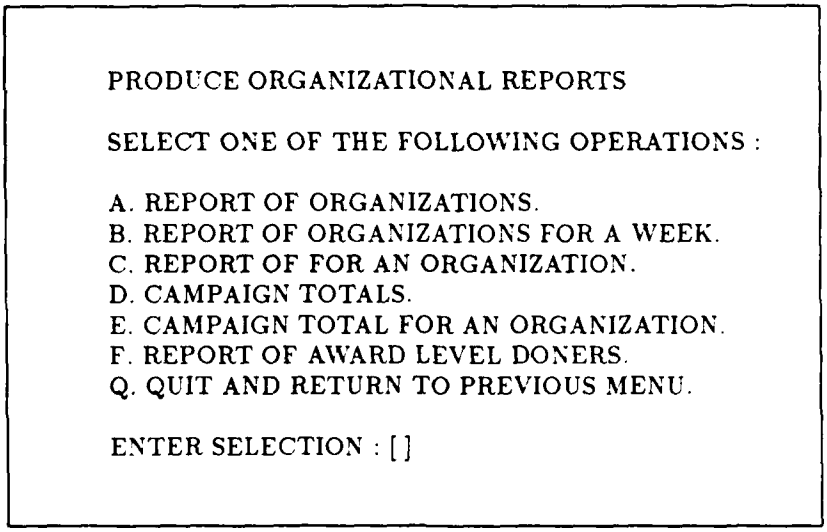


Figure 19. Option D.D : Produce Organization Reports

A. Report of Organizations : Organization summary reports for all organizations.

B. Report of Organizations for a Week : Organization summary reports for all organizations for a single week.

C. Report for an Organization : Organization summary report for a particular organization.

D. Campaign Totals : Report campaign status to date for all organizations.

E. Campaign Total for an Organization : Report campaign status for a particular organization.

F. Report of Award Level Donors : Report of award level donors for all organizations.

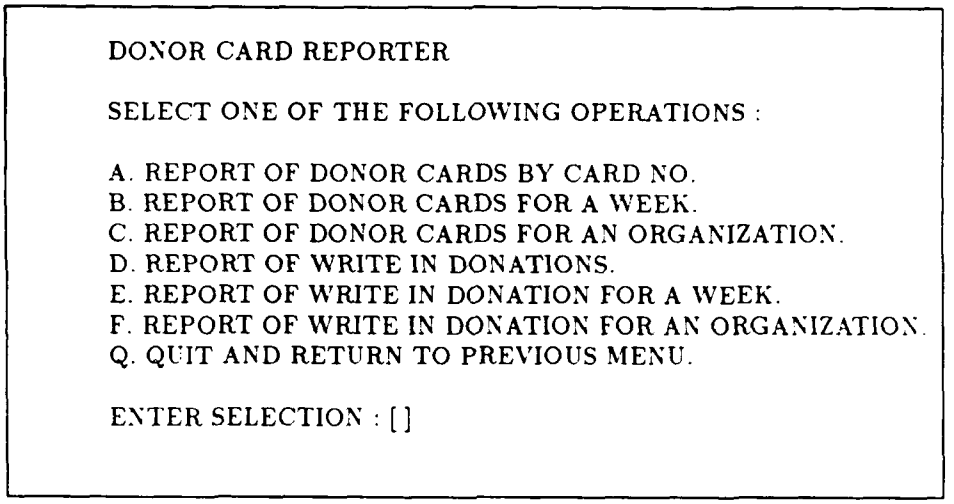


Figure 20. Option D.E : Donor Card Reports

A. Report of Donor Cards by Card No. : Produces a report of all donor cards by card number.

B. Report of Donor Cards for a Week : Produces a report of donor cards for a particular week.

C. Report of Donor Cards for an Organization : Produces a report of donor cards for a particular organization.

D. Report of Write in Donations : Produces a report of write in donations.

E. Report of Write in Donations for a Week : Produces a report of write in donations for a week.

F. Report of Write in Donations for an Organization : Produces a report of write in donations for an organization.

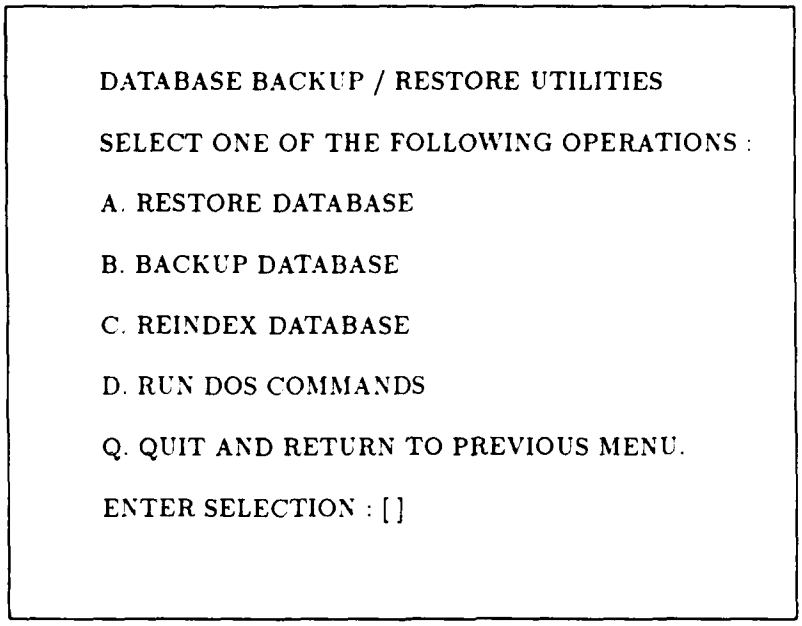


Figure 21. Option E : Backup/Restore Database

A. Restore Database : Restores the database from diskette and builds new indexs for all database files.

B. Backup Database : Backs up the database to diskette. The utility calculates the total number of diskettes needed to do the backup [8].

C. Reindex Database : Reindexs all database files. This option should be run if the database has any errors in reporting or editing.

D. Run DOS Commands : Allows the use of DOS commands. This option enables the formating of diskettes without leaving the system.

4.2.3 Coding. The coding was accomplished using the bottom-up design technique in which the lowest level modules were written to perform the loading of the database files, printing and reporting of the database files, writing the database to the ASCII file and other general utility functions. The donor card editor and error detection programs were then written. Each module was integrated into the menu structure for integration testing to insure compatability with the system as a whole.

4.2.4 Testing. The testing was performed in three phases. The first phase was when the modules were written. The modules were tested to insure they could perform the task they were written to perform. In the second phase the modules were added to the menu system and tested to ensure they performed properly. They were also tested for compatability with the system as a whole. The final phase was to run test data as a mini CFC campaign to check the system for ease of use and accuracy. The system was also tested for user acceptance by asking fellow students to run the system and give suggestions on the menus, messages and operation of the system.

V. *Implementation*

This chapter discussed issues on implementation of the CFC Collection System. The topics addressed are the issues in selecting a DBMS, file manipulation and constraints of DBASE III, and test results.

5.1 Database Selection

There is no clear cut solution in choosing the best DBMS. Every application has different problems and requirements to be fulfilled. When looking at a DBMS, the features of the DBMS must be weighed against the cost, performance, hardware requirements, availability of the software and many other factors [5].

Three DBMSs available for this project are DBASE III, ENABLE and CONDOR III. These three DBMSs are available to all government organizations on the small computer contract.

		DBASE III	CONDOR III	ENABLE
1	Max characters per record	4096	1024	64517
2	Max fields per record	128	127	254
3	Max records per table	unlimited	64517	65000
4	Max no. indexes per table	7	1	10
5	Max no. characters per index	254	127	100

Figure 22. DBMS Program Parameters

Five program parameters to be considered when evaluating the capability of a DBMS are listed in Figure 22 [5,6].

DBASE III with its unlimited maximum records per table is able to accommodate the largest number of data base records. The indexing capability of CONDOR III is not as flexible as the others. This may require additional programming and would be a limitation in some areas. CONDOR III also has the smallest capability as far as record size and quantity. ENABLE has the largest max characters per record and the largest number of indexes per table, however the seven indexes per table with DBASE III is more than adequate for this project. The record sizes in the CFC database will be small enough for any of the DBMS to handle.

Software Digest Inc., an independent organization that rates personal computer software, tested several DBMSs including DBASE III and CONDOR III. The overall evaluation stated DBASE III and CONDOR III received equal ratings. In error handling, ease of learning, and ease of use, DBASE III rated lower than CONDOR III. DBASE III rated higher than CONDOR III in the area of versatility and performance [7]. The area of versatility with DBASE III is high because of its programming language. However, the programming capability is also one of the reasons that DBASE III rates lower in the area of ease of learning. A powerful programming language is needed to build the CFC system so the end users will be using a set of menu driver programs. The

CFC system will provide an interface enabling the user to operate the system without a knowledge of how to use the DBMS it will run under. Thus, the added versatility and performance of DBASE III is highly desirable even though ease of use for the programmer will be diminished.

When selecting new software or hardware, previous work or compatibility of existing software or hardware must be considered. The Air University at Maxwell AFB has some programs written in DBASE II and has personnel knowledgeable in the use of DBASE III. If the same DBMS is used in the CFC system then less training will be required and maintenance will proceed more smoothly.

In consideration of the above factors we chose DBASE III for the CFC project.

5.2 File Manipulation and Constraints of DBASE III

DBASE III uses several different types of files. Two of the files will be discussed in this chapter. The first is database file. The database file provides for the orderly storage of data. The data is organized in records, which have a defined structure. The records are subdivided into fields of different types. When one file is related to another file, each file will have a common field. The common field enables the two files to be linked together forming the relationship as defined in the E-R diagram. The second file needed for the database structure is the index file. The index is built by using one or more of the fields in the database file. Indexes provide ordering of data and efficient access. A database file many have more than one index associated with it.

The database file used with an index enables DBASE III to locate records using the command **SEEK <expression>**. The **<expression>** in our case is the key of the record desired. The SEEK with an index is the fastest way to retrieve a record in a DBASE III file. The SEEK locates the record by searching the index for a match. When a match is found the associated database is positioned at the record containing the match. A not found indicator is set if no match is found in the index. Indexing and the SEEK command is used extensively in the CFC Collection System. One other important aspect of the index file is that index files are needed to order database files

to use the SET RELATION command. The SET RELATION command is used to align two or more database files together by a common field, making the one-to-one relation as described in Chapter III. To use SET RELATION one or both of the files have to be indexed. The many-to-many, one-to-many and many-to-one relations are accomplished by programming in DBASE III. The method used to do this is open all the files in the relationship. Using the SEEK command locate the primary record and read the necessary fields into memory. Using the fields necessary for the relation switch to the file which has the relation desired. This file is indexed on the fields in the relation and in ascending order. Using the SEEK command locate the first record in the relation, the rest are located with a while loop until the key field changes.

DBASE III proved to be a versitile tool for this project and all obstacles encountered could be worked out. The following were the problems encountered :

1. DBASE III limits the user to fifteen files open at one time. Ten of the open files can be database files. The fifteen files at first seems to be quite a bit. The problem is fifteen is the total number of files include index files, format files and program files. The database and index files were discussed in the previous section. The format files are the data entry screens. The program files are the actual programs.

   The file limit became a problem as soon as the donor card editor was incorporated into the menu system. The programs were orginally written as short modules which called other modules. Since DBASE III holds each module as an open file, the fifteen limit was quickly reached. The programs were restructured so a maximum level of called modules was three. This limit kept the system from exceeding the open file limit. However, the restructuring brought the line count of some of the modules to over five hundred lines.

2. The next constraint was the DBASE III Set Relation command. DBASE III allows the matching of two or more files with a common field by the use of the Set Relation command. The problem is that only a one-to-one relation can be accomplished with the Set Relation

command. The one-to-many and many-to-one relations have to be programmed, as in the case of the overflow and write in donation. This required extensive programming for the functions of the Donor Card Editor.

3. DBASE III does not navigate through several files quickly. When several files are operated on at the same time the system slows down significantly. This is noticable in two places, first when the total files for reports are built in the Campaign Report Section of the system and when copying donor cards from the temporary database to the actual database files. These two functions use several files at the same time. This causes the system to take extra time to switch back and forth between buffers while stepping through the files. These options are run in a batch mode, since the operator does not need to stay with the computer while they are running, the delay should not be too much of an inconvience.

5.3 Test Results

The testing was performed in three phases. The first phase was when the modules were written. The modules were tested to insure they could perform the task they were written to perform. In the second phase the modules were added to the menu system and tested to insure they performed properly. They were also tested for compatability with the system as a whole. The final phase was to run test data as a mini CFC campaign to check the system for ease of use and accuracy. The CFC system was tested in all three phases on a Z248 computer with a 8 mhz clock and also on an IBM PC compatible which has a 4.77 mhz clock. The test results showed the system fulfills the specified requirements.

The menu system provides a means to operate the CFC system in a short period of time. The donor card entry error checking ensures database integrity. The Z248 computer, having an 8 mhz clock, operates at a speed that makes the CFC system operate smoothly. The error checking slows the system down some what. If the system is used with a IBM PC compatible computer with a 4.77

mhz clock, the system would be acceptable for an office or keyworker at a small installation but would be too slow for a large installation. The error checking slows the system when the agencies, organizations and pay grades on the donor card are checked for valid numbers. This time delay remains constant throughout the campaign since these numbers do not change after the campaign begins.

VI. Conclusion and Future Requirements

This chapter summarizes what was accomplished in this project and also gives a suggestion for a future requirement to the system.

6.1 Summary

This thesis effort combines software engineering methods to design and implement the CFC Collection System. The database was designed using E-R modeling in which the required data elements were transformed into tables. These tables were implemented into the database files. The implementation of the database files was accomplished using the DBASE III DBMS.

DBASE III was chosen as the DBMS for the system because of its availability on the government small computer contract along with its power and versitility.

The software supporting the CFC Collection System was designed using the data-flow design method. The programs were written in the DBASE III programming language. By using the DBASE III programming language the entire CFC Collection System is supported by DBASE III without the need for outside language compilers. The only commerical software support needed is DBASE III.

The sponsor wanted a system to aid in the annual Combined Federal Campaign. The purpose of the system is to reduce the man-hours required to perform the necessary functions for the campaign. In addition to reducing the number of man hours required for the campaign, the system also improves the accuracy of reports generated by workers at different levels in the donating organizations. The audit trail is enhanced by the ability of the system to track donor cards by organization and week donated.

The CFC Collection System automates many parts of the annual Combined Federal Campaign. This system reduces the number of man hours required to support the campaign and

increases the accuracy and speed of the campaign process. The system accommodates the transfer of data between systems used on one installation.

6.2 Future Requirements

A future enhancement for the system might be a graphing tool for comparisons of previous campaigns and present status. An award program calculator could also be added for calculating gift levels. Because of the modular approach used in the software design, additional functions can be easily incorporated into the system.

Appendix A. *Database Files and Descriptions*

This appendix contains the file structures followed by a description of the files and their purpose.

Field	Field Name	Type	Length	Dec
1	CARD_NO	Numeric	5	
2	TYP	Character	1	
3	GRADE	Character	5	
4	GIFT_LEVEL	Character	1	
5	WEEK	Numeric	2	
6	CASH	Numeric	7	2
7	ALLOTMENT	Numeric	7	2
8	TOTAL	Numeric	7	2
9	SSAN	Numeric	9	
10	ORG_CODE	Numeric	2	3
11	SUB_CODE	Character	2	
12	WRITIN	Logical	1	
13	DESIGNATED	Logical	1	

Figure 23. Donation Card File (Doncard.dbf)

No	Index Name	Indexed by Fields
1	DONCD.ndx	CARD_NO
2	DONWK.ndx	WEEK
3	DOSWK.ndx	STR(ORG_CODE,3) + SUB_CODE + STR(WEEK,1)

Figure 24. Donation Card Index Files

Doncard : contains the fields from the donor card that are common to every donor card. ORG_CODE, SUB_CODE relates Doncard to Organ on a many-to-one basis. CARD_NO relates Doncard to Person on a one-to-one basis and to Writin on a one-to-many basis. Agency is related to Doncard via Givento and Ogivento on a many-to-many basis. The common field in Givento and Ogivento is CARD_NO for Doncard and AGENCY_NO for Agency.

The uses for the index files are as follows:

1. Doncd : This index in the main index and is used in donor card editing and running the campaign. Doncard is not permitted to have duplicate donations with the same card number.
2. Donwk : This index is used in reporting week totals in the Donor Card Reports.
3. Doswk :This index is used in reporting where the organizations are totaled by organization and by week.

Field	Field Name	Type	Length	Dec
1	CARD_NO	Numeric	5	
2	AGENCY_1	Numeric	4	
3	AMOUNT_1	Numeric	7	2
4	AGENCY_2	Numeric	4	
5	AMOUNT_2	Numeric	7	2
6	AGENCY_3	Numeric	4	
7	AMOUNT_3	Numeric	7	2
8	AGENCY_4	Numeric	4	
9	AMOUNT_4	Numeric	7	2
10	AGENCY_5	Numeric	4	
11	AMOUNT_5	Numeric	7	2
12	OVERFLOW	Logical	1	

Figure 25. Designated Gift File (Givento.dbf)

No	Index Name	Indexed by Fields
1	GIVCD.ndx	CARD_NO

Figure 26. Designated Gift Index Files

Givento : This file holds five agencies and an amount for each agency. Givento is used when an agency is designated on the donor card and is utilized fairly often. One index file is used for Givento. Agency is related to Doncard via Givento by AGENCY_NO and CARD_NO on a many-to-many basis, The limit per card is 5 agencies.

Field	Field Name	Type	Length	Dec
1	CARD_NO	Numeric	5	
2	AGENCY_1	Numeric	4	
3	AMOUNT_1	Numeric	7	2
4	AGENCY_2	Numeric	4	
5	AMOUNT_2	Numeric	7	2
6	AGENCY_3	Numeric	4	
7	AMOUNT_3	Numeric	7	2
8	AGENCY_4	Numeric	4	
9	AMOUNT_4	Numeric	7	2
10	AGENCY_5	Numeric	4	
11	AMOUNT_5	Numeric	7	2
12	OVFLO_NO	Numeric	2	

Figure 27. Overflow Card File (Ogivento.dbf)

No	Index Name	Indexed by Fields
1	OGIVCD.ndx	CARD_NO

Figure 28. Overflow Card Index Files

Ogivento : This file holds five agencies and an amount for each. This file is used when a donor wants to specify more than five agencies on a donation card. These donation cards are referred to as overflow cards. This file is rarely used. Most donors do not designate more than five agencies. The extra file saves space in the database and adds flexibility for the donor card entry. One index file is used for Ogivento. Agency is related to Doncard by Ogivento by AGENCY_NO and CARD_NO on a many-to-many basis. The limit per card is 5 agencies.

Field	Field Name	Type	Length	Dec
1	CARD_NO	Numeric	5	
2	AGENCY_NAM	Character	60	
3	ADDRESS	Character	35	
4	CITY	Character	20	
5	STATE	Character	2	
6	ZIP	Numeric	5	
7	ANNUAL_AMT	Numeric	7	2
8	WRIT_NO	Numeric	2	

Figure 29. Write in Card File (Writin.dbf)

No	Index Name	Indexed by Fields
1	WRITCD.ndx	CARD_NO

Figure 30. Write in Card Index Files

Writin : The write in file allows a donor to specify an agency that is not in the CFC catalog. The write in file is related to Doncard by CARD_NO on a many-to-one basis allowing a donor to specify as many write in donations as he or she wants. The ability to handle write in agencies is necessary. People often write in the agency they want their donation to go to. The extra file enables the system to handle the write in agencies and also to report on them so they can be added to the CFC catalog for the next next campaign. One index file is used for Writin.

Field	Field Name	Type	Length	Dec
1	ORG_CODE	Numeric	3	
2	SUB_CODE	Character	2	
3	ORG_NAME	Character	35	
4	ADDRESS	Character	35	
5	CITY	Character	20	
6	STATE	Character	2	
7	ZIP	Numeric	5	
8	COMMANDER	Character	20	
9	COM_PHONE	Character	14	
10	PROJ_OFF	Numeric	20	
11	PO_PHONE	Numeric	14	
12	NO_PERSON	Numeric	6	
13	GOAL	Numeric	11	2
14	PGOAL	Numeric	11	2
15	PORG_CODE	Numeric	3	
16	PSUB_CODE	Character	2	
17	HNO_PERSON	Numeric	6	
18	HGOAL	Numeric	11	2

Figure 31. Organization File (Organ.dbf)

No	Index Name	Indexed by Fields
1	ORGNO.ndx	STR(ORG_CODE,3) + SUB_CODE
2	ORGR.ndx	as required by report

Figure 32. Organization Index Files

Organ is the organizational information file. This file is used to hold and calculate goal information and for error checking donations for valid organizations. Organ is related by ORG_CODE and SUB_CODE to person on a one-to-many basis and also to Doncard on a one-to-many basis.

The uses for the indexed are as follows:

1. Orgno : This is the main index for the organization file and is used to locate organizations for error checking during donor card entry, reporting on organizations ordered by organization code and sub code and also editing and entry.
2. Orgr : indexed on several fields and is used only for reports.

Field	Field Name	Type	Length	Dec
1	PAY_GRADE	Character	5	
2	SUGG_GIFT	Numeric	8	2

Figure 33. Suggest Gift File (Suggift.dbf)

No	Index Name	Indexed by Fields
1	SUGPG.ndx	PAY_GRADE

Figure 34. Suggest Gift Index Files

Suggift : This is the suggested gift file. This file holds suggested gift information and is used for goal projections and error checking donation cards for proper pay grades when an allotment is used for the donation. One index file is used for the suggested gift.

Field	Field Name	Type	Length	Dec
1	SSAN	Numeric	9	
2	LNAME	Character	10	
3	FINIT	Character	1	
4	TYP	Character	1	
5	SERVICE	Character	5	
6	GRADE	Character	5	
7	ORG_CODE	Numeric	3	
8	SUB_CODE	Character	2	
9	PHONE_NO	Character	14	
10	KEYWORKER	Logical	1	
11	CARD_NO	Numeric	5	

Figure 35. Personnel File (Person.dbf)

No	Index Name	Indexed by Fields
1	PERSS.ndx	SSAN
2	PERCD.ndx	CARD_NO
3	PERR.ndx	as required by report

Figure 36. Personnel Index Files

Person : This file holds information on personnel. Personnel information is needed when the donor makes a donation by allotment or wants to have his or her donation recorded. This would be used for an award program which sends out acknowledgements of the donations which are at certain amounts. The personnel file is also used to keep a record of keyworkers. Error checking uses this file to for check duplicate donations per person. Reporting of donations for military personnel and civilians also uses personnel information. Person is related to Doncard by CARD_NO on a one-to-one basis and to Organ by ORG_CODE and SUB_CODE on a many-to-one basis.

The uses for the index files are as follows:

1. Perss : This index is used to edit personnel file and also to report on personnel by SSAN.
2. Percd : This index is used so the SEEK verb can locate a person by CARD_NO.
3. Perr : indexed on several fields and is used only for reports.

Field	Field Name	Type	Length	Dec
1	AGENCY_NO	Numeric	4	
2	AGENCY_NAM	Character	60	
3	ADDRESS	Character	35	
4	CITY	Character	20	
5	STATE	Character	2	
6	ZIP	Numeric	5	
7	TELE	Character	14	
8	DESCRIP	Character	240	
9	PAGENCY_NO	Numeric	4	

Figure 37. Agency File (Agency.dbf)

No	Index Name	Indexed by Fields
1	AGNO.ndx	AGENCY_NO
2	AGR.ndx	as required by report

Figure 38. Agency Index Files

Agency : contains valid agencies for the CFC campaign. The agency file is used for error checking donor cards for valid agencies when the donor designates specific agencies to receive all or part or his or her donation as shown. Agency is related to Doncard via Givento and Ogivento. The common fields are AGENCY_NO for Agency and CARD_NO for Doncard.

The uses for the index files are as follows:

1. Agno : This is the main index for the Agency file and is used to located agencies for error checking during donor card entry, reporting on agencies ordered by agency number and agency editing and entry.

2. Agr : indexed on several fields and is used only for reports.

Field	Field Name	Type	Length	Dec
1	AGENCY_NO	Numeric	4	
2	WK1	Numeric	9	2
3	WK1	Numeric	9	2
4	WK3	Numeric	9	2
5	WK4	Numeric	9	2
6	WK5	Numeric	9	2
7	WK6	Numeric	9	2
8	WK7	Numeric	9	2
9	TOTAL	Numeric	10	2

Figure 39. Agency Totals File (Agtot.dbf)

No	Index Name	Indexed by Fields
1	AGTOT.ndx	AGENCY_NO

Figure 40. Agency Totals Index Files

Agtot: holds totals by week and the total for the campaign of all designated donations and is use for agency reports. One index file is used for Agtot.

Field	Field Name	Type	Length	Dec
1	ORG_CODE	Numeric	3	
2	SUB_CODE	Character	2	
3	WEEK	Numeric	1	
4	CONTRIB	Numeric	5	
5	DED_TOT	Numeric	11	2
6	PAY_DED	Numeric	5	
7	PAID_NOW	Numeric	11	2
8	TOTAL	Numeric	11	2
9	SUG_GIVER	Numeric	5	
10	UNDESTOT	Numeric	11	2
11	DESTOT	Numeric	11	2
12	MILTOT	Numeric	11	2
13	NOMIL	Numeric	5	
14	OFFTOT	Numeric	11	2
15	NOOFF	Numeric	5	
16	ENLTOT	Numeric	11	2
17	NOENL	Numeric	5	
18	CIVTOT	Numeric	11	2
19	NOCIV	Numeric	5	

Figure 41. Organization Totals File (Orgtot.dbf)

No	Index Name	Indexed by Fields
1	ORGTOT.ndx	STR(ORG_CODE,3) + SUB_CODE + STR(WEEK,1)
2	ORGWK.ndx	WEEK
3	ORGR.ndx	as report requires

Figure 42. Organization Totals Index Files

Orgtot: contains summary report information for all organizations in the database and is used for campaign summary and historical reporting.

The uses for the index files are as follows:

1. Orgtot : This index is used in reporting where the organizations are reported by organization and by week.
2. Orgwk : This index is used in reporting week totals for a particular week in the campaign.
3. Orgr : indexed on several fields and is used only for reports.

Field	Field Name	Type	Length	Dec
1	ORG_CODE	Numeric	3	
2	SUB_CODE	Character	2	
3	CONTRIB	Numeric	5	
4	PER_PART	Numeric	3	
5	TOTAL	Numeric	11	2
6	PER_GOAL	Numeric	3	
7	SUG_GIVER	Numeric	5	

Figure 43. Campaign Totals File (Camtot.dbf)

No	Index Name	Indexed by Fields
1	CAMTOT.ndx	STR(ORG_CODE,3) + SUB_CODE
2	CAMR.ndx	as report requires

Figure 44. Campaign Totals Index File

Camtot : contains campaign total information for all organizations in the database and is used for campaign total reports and historical reports.

The uses for the index files are as follows:

1. Camtot : This index is used in reporting where the organizations are reported by organization and by week.
2. Camr : indexed on several fields and is used only for reports.

The temporary and duplicate files support the loading and unloading of the database files.

Field	Field Name	Type	Length	Dec
1	CARD_NO	Numeric	5	
2	SSAN	Numeric	9	
3	LNAME	Character	10	
4	FINIT	Character	1	
5	ORG_CODE	Numeric	2	3
6	SUB_CODE	Character	2	
7	WEEK	Numeric	2	
8	TYP	Character	1	
9	SERVICE	Character	5	
10	GRADE	Character	5	
11	GIFT_LEVEL	Character	1	
12	CASH	Numeric	7	2
13	ALLOTMENT	Numeric	7	2
14	TOTAL	Numeric	7	2
15	AGENCY_1	Numeric	4	
16	AMOUNT_1	Numeric	7	2
17	AGENCY_2	Numeric	4	
18	AMOUNT_2	Numeric	7	2
19	AGENCY_3	Numeric	4	
20	AMOUNT_3	Numeric	7	2
21	AGENCY_4	Numeric	4	
22	AMOUNT_4	Numeric	7	2
23	AGENCY_5	Numeric	4	
24	AMOUNT_5	Numeric	7	2

Figure 45. Temporary Donor Card File (Tempdon.dbf)

No	Index Name	Indexed by Fields
1	TDONCD.ndx	CARD_NO

Figure 46. Temporary Donor Card Index Files

Field	Field Name	Type	Length	Dec
1	SSAN	Numeric	9	
2	LNAME	Character	10	
3	FINIT	Character	1	
4	TYP	Character	1	
5	SERVICE	Character	5	
6	GRADE	Character	5	
7	ORG_CODE	Numeric	3	
8	SUB_CODE	Character	2	
9	PHONE_NO	Character	14	
10	KEYWORKER	Logical	1	
11	CARD_NO	Numeric	5	

Figure 51. Temporary Personnel File (Tempper.dbf)

No	Index Name	Indexed by Fields
1	TPERSS.ndx	SSAN

Figure 52. Temporary Personnel Index Files

Field	Field Name	Type	Length	Dec
1	AGENCY_NO	Numeric	4	
2	AGENCY_NAM	Character	60	
3	ADDRESS	Character	35	
4	CITY	Character	20	
5	STATE	Character	2	
6	ZIP	Numeric	5	
7	TELE	Character	14	
8	DESCRIP	Character	240	
9	PAGENCY_NO	Numeric	4	

Figure 53. Temporary Agency File (Tempag.dbf)

No	Index Name	Indexed by Fields
1	TAGNO.ndx	AGENCY_NO

Figure 54. Temporary Agency Index Files

Field	Field Name	Type	Length	Dec
1	ORG_CODE	Numeric	3	
2	SUB_CODE	Character	2	
3	ORG_NAME	Character	35	
4	ADDRESS	Character	35	
5	CITY	Character	20	
6	STATE	Character	2	
7	ZIP	Numeric	5	
8	COMMANDER	Character	20	
9	COM_PHONE	Character	14	
10	PROJ_OFF	Numeric	20	
11	PO_PHONE	Numeric	14	
12	NO_PERSON	Numeric	6	
13	GOAL	Numeric	11	2
14	PGOAL	Numeric	11	2
15	PORG_CODE	Numeric	3	
16	PSUB_CODE	Character	2	
17	HNO_PERSON	Numeric	6	
18	HGOAL	Numeric	11	2

Figure 47. Temporary Organization File (Temporg.dbf)

No	Index Name	Indexed by Fields
3	TORGNO.ndx	STR(ORG_CODE,3) + SUB_CODE

Figure 48. Temporary Organization Index Files

Field	Field Name	Type	Length	Dec
1	PAY_GRADE	Character	5	
2	SUGG_GIFT	Numeric	8	2

Figure 49. Temporary Suggest Gift File (Tempsug.dbf)

No	Index Name	Indexed by Fields
1	TSUGPG.ndx	PAY_GRADE

Figure 50. Temporary Suggest Gift Index Files

Field	Field Name	Type	Length	Dec
1	CARD_NO	Numeric	5	
2	SSAN	Numeric	9	
3	LNAME	Character	10	
4	FINIT	Character	1	
5	ORG_CODE	Character	3	
6	SUB_CODE	Character	2	
7	WEEK	Character	1	
8	TYP	Character	1	
9	SERVICE	Character	5	
10	GRADE	Character	5	
11	GIFT_LEVEL	Character	1	
12	CASH	Character	8	
13	ALLOTMENT	Character	8	
14	TOTAL	Character	8	
15	AGENCY_1	Character	4	
16	AMOUNT_1	Character	8	
17	AGENCY_2	Character	4	
18	AMOUNT_2	Character	8	
19	AGENCY_3	Character	4	
20	AMOUNT_3	Character	8	
21	AGENCY_4	Character	4	
22	AMOUNT_4	Character	8	
23	AGENCY_5	Character	4	
24	AMOUNT_5	Character	8	

Figure 55. Duplicate Donor Card (Dupdon.dbf)

No	Index Name	Indexed by Fields
1	DDONCD.ndx	CARD_NO

Figure 56. Duplicate Donor Card Index Files

Field	Field Name	Type	Length	Dec
1	ORG_CODE	Numeric	3	
2	SUB_CODE	Character	2	
3	ORG_NAME	Character	35	
4	ADDRESS	Character	35	
5	CITY	Character	20	
6	STATE	Character	2	
7	ZIP	Numeric	5	
8	COMMANDER	Character	20	
9	COM_PHONE	Character	14	
10	PROJ_OFF	Numeric	20	
11	PO_PHONE	Numeric	14	
12	NO_PERSON	Numeric	6	
13	GOAL	Numeric	11	2
14	PGOAL	Numeric	11	2
15	PORG_CODE	Numeric	3	
16	PSUB_CODE	Character	2	
17	HNO_PERSON	Numeric	6	
18	HGOAL	Numeric	11	2

Figure 57. Duplicate Organization File (Duporg.dbf)

No	Index Name	Indexed by Fields
1	DORGNO.ndx	STR(ORG_CODE,3) + SUB_CODE

Figure 58. Duplicate Organization Index Files

Field	Field Name	Type	Length	Dec
1	PAY_GRADE	Character	5	
2	SUGG_GIFT	Numeric	8	2

Figure 59. Duplicate Suggested Gift File (Dupsug.dbf)

No	Index Name	Indexed by Fields
1	DSUGPG.ndx	PAY_GRADE

Figure 60. Duplicate Suggested Gift Index Files

Field	Field Name	Type	Length	Dec
1	SSAN	Numeric	9	
2	LNAME	Character	10	
3	FINIT	Character	1	
4	TYP	Character	1	
5	SERVICE	Character	5	
6	GRADE	Character	5	
7	ORG_CODE	Numeric	3	
8	SUB_CODE	Character	2	
9	PHONE_NO	Character	14	
10	KEYWORKER	Logical	1	
11	CARD_NO	Numeric	5	

Figure 61. Duplicate Personnel File (Dupper.dbf)

No	Index Name	Indexed by Fields
1	DPERSS.ndx	SSAN

Figure 62. Duplicate Personnel Index Files

Field	Field Name	Type	Length	Dec
1	AGENCY_NO	Numeric	4	
2	AGENCY_NAM	Character	60	
3	ADDRESS	Character	35	
4	CITY	Character	20	
5	STATE	Character	2	
6	ZIP	Numeric	5	
7	TELE	Character	14	
8	DESCRIP	Character	240	
9	PAGENCY_NO	Numeric	4	

Figure 63. Duplicate Agency File (Dupag.dbf)

No	Index Name	Indexed by Fields
1	DAGNO.ndx	AGENCY_NO

Figure 64. Duplicate Agency Index Files

Appendix B. *System Programs*

This appendix contains tables of the system programs. The tables contain the calling program, the programs called by the program and their purpose.

Called by	Calls	Function
	AGMENU	Agency Utilities : menu
	AGSRPT	Campaign Reports : Agency campaign reports
	BAKUP	Backup/Restore Database: Backup the database to diskette
	BALCALC	Campaign Reports : Build Summary Files
	DONCPROC	Donation Utilities : Copy Temporary to Database
	DONCRPT	Donation Utilities : Report on Donor Cards
	DONCTOTP	Donation Utilities : Copy Database to ASCII File
	DONEDIT	Donation Utilities : Edit/Review Files
	DONREAD	Donation Utilities : Load ASCII File to Temporary
	DONRPROC	Donation Utilities : Report on Files
	DONSET	Donation Utilities : sets defaults for doner card
	ERRORS	Donation Utilities : error codes for copying temporary to donation database files
	GOALMENU	Organization Goals : menu
	HISRPT	Campaign Reports : Historical campaign Reports
	ORGMENU	Organization Utilities : menu
	ORGRPT	Campaign Reports : Organization campaign Reports
	PERMENU	Personnel Utilities : menu
	RDOS	Backup/Restore Database : Format Diskettes, dos window
	RESTR	Backup/Restore Database : Restore Database
	SUGMENU	Suggested Gift Utilities : menu
	TDONEDIT	Donation Utilities : Edit/Review Temporary, Duplicates

Figure 65. Main

Called by	Calls	Function
MAIN	AGCOPY	Agency Utilities : Copy Database to ASCII File
	AGEPROC	Agency Utilities : Edit/Review Files
	AGREAD	Agency Utilities : Load ASCII File to Temporary
	AGRPROC	Agency Utilities : Report on Files

Figure 66. Agmenu

Called by	Calls	Function
MAIN	DONAPP	Donation Utilities : append new doner card
	DONCCK	Donation Utilities : error check doner card
	DONDEL	Donation Utilities : delete doner card
	DONREP	Donation Utilities : add or replace Doncard
	DONUP	Donation Utilities : read donation into memory
	GIVREP	Donation Utilities : add or replace Givento
	OVFEDIT	Donation Utilities : add or edit overflow cards
	PERREP	Donation Utilities : add or replace Person
	WRITEDIT	Donation Utilities : add or edit write in cards

Figure 67. Donedit

Called by	Calls	Function
MAIN	GOALADJ	Organization Goals : Organization Goal Adjusting
	GOALCALC	Organization Goals : Organziation Heirarchy
	GOALPRO	Organization Goals : Individual Organizations
	ORGMENU	Organization Utilities : menu

Figure 68. Goalmenu

Called by	Calls	Function
MAIN	ORGCOPY	Organization Util : Copy Temporary to Database
GOALMENU	ORGEPROC	Organization Util : Edit/Review Files
	ORGREAD	Organization Util : Load ASCII File to Temporary
	ORGRPROC	Organization Util : Report on Files

Figure 69. Orgmenu

Called by	Calls	Function
DONEDIT	OVFADD	Donation Utilities : add overflow card
	OVFAGCK	Donation Utilities : agency check overflow card
	OVFREP	Donation Utilities : replace overflow card
	OVFUP	Donation Utilities : read Ogivento into memory

Figure 70. Ovfedit

Called by	Calls	Function
MAIN	PERCOPY	Personnel Utilities : Copy Temporary to Database
	PEREPROC	Personnel Utilities : Edit/Review Files
	PERREAD	Personnel Utilities : Load ASCII File to Temporary
	PERRPROC	Personnel Utilities : Report on Files

Figure 71. Permenu

Called by	Calls	Function
MAIN	SUGCOPY	Suggested Gift Utilities : Copy Temporary to Database
	SUGEPROC	Suggested Gift Utilities : Edit/Review Files
	SUGREAD	Suggested Gift Utilities : Load ASCII File to Temporary
	SUGRPROC	Suggested Gift Utilities : Report on Files

Figure 72. Sugmenu

Called by	Calls	Function
DONEDIT	WRITAPP	Donation Utilities : add write in agency
	WRITCK	Donation Utilities : error check donation
	WRITREP	Donation Utilities : replace write in agency
	WRITUP	Donation Utilities : read Writin into memory

Figure 73. Writedit

Appendix C. *Format Screens and Report Forms*

This appendix contains tables containing the program using the screen or report form, the name of the screen or report form and the function of the screen or report form. The format screen is the data entry screen. The report form are the reports generated by the DBASE III report form generator.

Screens	Used by	Function
AGFND	AGEPROC	Agency lookup screen
AGFRM	AGEPROC	Agency edit/review screen
AGLFRM	AGSRRPT	Agency label printing screen
DONAGFX	DONEPROC, OVAGCK	Donor card error screen
DONCFX	DONCLR	Donor card error screen
DONFND	DONEDIT	Donor card editor entry screen
DONFRM	DONADD, DONUP	Donor card edit/review screen
GOALCFRM	GOALPRO	Goal Calculator entry Screen
GOALFND	GOALPRO	Find organization for goal entry
ORGFND	ORGEPROC	Organization editor entry screen
ORGFRM	ORGEPROC	Organization edit/review screen
ORGLFRM	ORGRPT	Organization label printing screen
ORGRFND	DONCPROC, ORGRPT	Find organization for donor card report
ORGRFRM	ORGRPROC	Find Organziation for organization report
OVFFRM	OVFADD, OVFUP	Overflow card edit/review screen
PERFRM	PEREDIT	Personnel edit/review screen
PERRFRM	PERRPROC	Find organization for personnel report
SETFRM	DONSET	Screen to change default for donor card editor
SUGFRM	SUGEPROC	Screen for suggest gift edit/review
TAGFRM	AGEPROC	Temporary and Duplicate Agency edit/review
TDONFRM	TDONEDIT	Donor card temporary and duplicate edit/review
TORGFRM	ORGEPROC	Organization temporary and duplicate edit/review
TPERFRM	PEREPROC	Personnel temporary and duplicated edit/review
TSUGFRM	SUGEPROC	Suggest Gift temporary and duplicate edit/review
WRTFRM	WRITUP	Write in editor edit/review screen
WRTFX	WRITFX	Write in error screen

Figure 74. Format Files

Report Form	Used By	Function
AGTOT	AGSRPT	Agency totals
AGDOF	AGSRPT	Agency distribution of funds
AGLAB	AGSRPT	Agency mailing Labels
AWARDRPT	ORGRPT	Award report for giving
CAMTOT	ORGRPT	Campaign Totals
CAMOTOT	ORGRPT	Campaign totals for an organization
CAMWTOT	ORGRPT	Campaign total for a particular week
DONORPT	DONCRPT	Donor card report for an organization
DONRPT	DONCRPT	Donor card report for campaign
DONWRPT	DONCRPT	Donor card report for a particular week
DORGRPT	ORGRPROC	Duplicate organization report
DPERRPT	PERRPROC	Duplicate personnel report
DSUGRPT	SUGRPROC	Duplicate suggest gift report
HISOTOT	HISRPT	Historical report for an organization
HISTOT	HISRPT	Historical report for a campaign
HISWTOT	HISRPT	Historical report for a particular week
ORGBAL	ORGRPT	Organization balance report
ORGCLAB	HISRPT	Organization mailing labels to commanders
ORGCPRPT	HISRPT	Organization campaign proposal report
ORGCSRPT	HISRPT	Organization campaign collection sheet
ORGPOLAB	HISRPT	Organization mailing label to project officer
ORGOTOT	ORGRPT	Organization totals for an organization
ORGPRPT	ORGRPROC	Organization progress report
ORGRPT	ORGRPROC	Organization report
ORGTOT	ORGRPT	Organization totals
ORGWTOT	ORGRPT	Organization totals for a particular week
PERORPT	PERRPROC	Personnel report for an organiation
PERRPT	PERRPROC	Personnel report
SUGCRPT	SUGRPROC	Suggested gift report for civilians
SUGMRPT	SUGRPROC	Suggested gift report for military
SUGRPT	SUGRPROC	Suggest gift report
TORGRPT	ORGRPROC	Temporary organization report
TPERRPT	PERRPROC	Temporary personnel report
TSUGRPT	SUGRPROC	Temporary suggest gift report
WRTORPT	DONCPRT	Write in report for an organization
WRTRPT	DONCPRT	Write in report by card number
WRTWRPT	DONCPRT	Write in report for a particular week

Figure 75. Report Forms

Bibliography

1. Watkins, Richard E. *Development of Database System and Software to Support the Combined Federal Campaign.* Proposed AFIT Thesis Topic EE-5.
2. *Leading Edge NUTSHELL Information Filer.* Leading Edge Software Products Inc. Needham Heights, MA, 1985.
3. Pressman, Roger S. *Software Engineering: A Practitioner's Approach.* McGraw-Hill. New York, 1982.
4. Korth, Henry F. Silberilberschaltz, Abraham. *Database System Concepts.* McGraw-Hill. New York, 1986.
5. Davidson, Dennis. *A Decision Support System for Database Software Evaluation.* Master's thesis, Air Force Institute of Technology, 1986.
6. Zenith Data Systems. ENABLE Database Manager. Zenith Data Systems, ST Joseph, MO, 1985.
7. Software Digest Rating Newsletter. *Relational Database Management Programs for IBM, AT&T, Compaq and Other Compatible Personal Computers.* April 1985.
8. Liskin, Miriam. *Advanced dBASE III Plus Programming And Techniques.* Osborne McGraw-Hill Berkeley, California, 1987.

Vita

Captain Frank L. Ucman was born on 26 September, 1951 in Calument, Michigan. He graduated from Calumet High School in Calumet, Michigan in 1969. He graduated from the Troy State University at Troy, Alabama in June, 1982 with a Bachelor of Science degree in Computer Science and received his commission in the USAF. After receiving his commission he served in a database shop at SAC Headquarters Offutt AFB, Nebraska until May, 1986 at which time he entered the Air Force Institute of Technology.

Permanent address: 127 N. Pewabic St.
Laurium, Mich 49913

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; Distribution Unlimited
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GCS/ENG/87D-27	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION School of Engineering 6b. OFFICE SYMBOL (If applicable) AFIT/ENG	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB, OH 45433-6583	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING / SPONSORING ORGANIZATION DCS Comptroller 8b. OFFICE SYMBOL (If applicable) AU/AC	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code) Maxwell AFB, AL 36112-5001	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

DESIGN AND IMPLEMENTATION OF THE COMBINED FEDERAL CAMPAIGN COLLECTION SYSTEM

12. PERSONAL AUTHOR(S)

Frank L. Ucman, B.S., Captain, US Air Force

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
MS Thesis	FROM _____ TO _____		82

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	
09	02		Combined Federal Campaign, DBASE III, Entity-Relationship Model, Database Management Systems

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Thesis Chairman: Mark A. Roth, Captain, USAF
Assistant Professor of Computer Systems.

Approved for public release: IAW AFR 190-1.
21 May 88
Dean for Research and Professional Development

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Capt Mark A. Roth	22b. TELEPHONE (Include Area Code) (513) 255-5533 22c. OFFICE SYMBOL AFIT/ENG

DD Form 1473, JUN 86 Previous editions are obsolete. SECURITY CLASSIFICATION OF THIS PAGE

Each year, the Department of Defense and other federal agencies participate in the Combined Federal Campaign (CFC). This campaign consolidates fund raising for many worthwhile charities in a single effort. Computer automation for the annual CFC drive would reduce the number of manhours required to support the annual campaign.

The CFC Collection System developed in this thesis automates many parts of the annual Combined Federal Campaign. The system is menu driven for ease of use and has the capability to establish fund raising goals and produces reports on contributions, agencies recieving money, donating organizations, and historical comparisons. The system reduces the number of man hours required to support the campaign and increases the accuracy and speed of the campaign process.

This thesis effort combined software engineering and database design methods to design and implement the CFC Collection System. The database was designed using the Entity Relationship (E-R) model. The E-R model identified the required data elements and the relationships between elements. The E-R model was then translated into a relational model, producing the tables required for implementation. The implementation of the database files was accomplished using the DBASE III database management system. The application software was designed using the data flow oriented approach to software design. The programs are written using a structured design and implemented in the DBASE III programming language.

END
DATED
FILM
8-88
DTIC